

# **EXHIBIT 22**

**IN THE UNITED STATES DISTRICT COURT  
FOR THE EASTERN DISTRICT OF TEXAS  
MARSHALL DIVISION**

EXPRESS MOBILE LLC;

Plaintiff,

v.

SVANACO, INC.;

Defendant.

Civil Action No. 2:17-cv-00130-JRG

Lead Case

---

BIGCOMMERCE, INC.

Defendant.

---

Civil Action No. 2:17-cv-00160-JRG

**INVALIDITY CONTENTIONS OF DEFENDANT BIGCOMMERCE, INC.**

Pursuant to Patent Local Rule 3-3, Defendant BigCommerce, Inc. (“BigCommerce”) makes the following disclosure of invalidity contentions to Plaintiff Express Mobile, Inc. (“Express Mobile”) regarding the asserted claims of U.S. Patent Nos. 6,546,397 and 7,594,168 (“’397 and ’168 patents”).

This preliminary disclosure is based on BigCommerce’s present understanding of Express Mobile’s interpretation of the claims of the ’397 and ’168 patents as advanced by Express Mobile in its Disclosure of Asserted Claims and Infringement Contentions. Nothing in BigCommerce’s disclosures should be regarded as necessarily reflecting the proper interpretation of the claims or an interpretation of the claims BigCommerce agrees with or proposes. BigCommerce disputes Express Mobile’s apparent interpretations and will propose alternative constructions at the appropriate time in the case.

BigCommerce's investigation regarding prior art grounds of invalidity is ongoing.

BigCommerce reserves the right to amend this disclosure consistent with Patent Rule 3-6 or otherwise as the Court may allow.

# **I. DISCLOSURES UNDER PATENT 3-3(a)**

The following list provides the identity of each item of prior art, patent, or publication that anticipates or renders obvious each asserted claim of the patents-in-suit.

## **A. Prior Art Patents**

<b>Patent Number</b>	<b>Country</b>	<b>Filing/Priority Date</b>	<b>Date of Issue/Publication</b>	<b>Short Cite</b>
6,313,835	US	4/9/1999	11/6/2001	Gever
6,219,680	US	6/19/1997	4/17/2001	Bernardo Ecommerce
6,185,587	US	6/19/1997	2/6/2001	Bernardo Help
5,940,834	US	3/13/1997	8/17/1999	Pinard

## **B. Prior Art Publications**

<b>Title</b>	<b>Date of Publication</b>	<b>Author</b>	<b>Short Cite</b>
SilverStream	February 14, 1998	SilverStream.com	SilverStream
Creating dynamic WWW pages by demonstration	May 1997	Robert Miller, Brad Myers	Miller
Responsive interaction for a large web application	April 1997	Crespo et al.	WebWriter II or WWII
WebWriter: a browser-based editor for constructing web applications	May 1996	Crespo et al.	WebWriter

Creating Geocities websites	March 1, 1999	Geocities	Geocities
Geocities S-1	June 1998	Geocities	Geocities S-1

### C. Prior Art Systems or Services

System/Service	Relevant Date	Persons/Entities Involved in Prior Use, Sale, and/or Offers for Sale	Short Cite
SilverStream	At least 1997	SilverStream, David Skok (corroborating patent application US 2002/0091725A1 which was subsequently abandoned).	SilverStream
WebWriter I, II	At least 1997	Andrew Crespo, Eric Bier	WWI, WWII, or WebWriter I, and WebWriter II
Geocities	By 1998	Geocities offered a community of personal websites. In the fall of 1998, Geocities launched a WYSIWYG web-page builder. The use of databases is corroborated by Geocities S-1	Geocities; Geocities S-1

## II. DISCLOSURE UNDER PATENT RULE 3-3(b)

BigCommerce hereby identifies each item of prior art that anticipates each asserted claim of the '397 and '168 patents under 35 U.S.C. 102(a), (b), (e), and (g). In addition, BigCommerce identifies prior art that renders each asserted claim obvious.

### A. Anticipatory Prior Art

Claims 1-6, 8, 9, 11, and 37 of the '397 patent and claims 1 and 6 of the '168 patent are anticipated by at least: Gever, Bernardo Ecommerce, Bernardo Help, Pinard, SilverStream, Miller, WebWriter I, WebWriter II, and Geocities.

## **B. Obviousness Prior Art**

BigCommerce contends that each reference identified above, when combined with the knowledge of persons of ordinary skill in the art at the time the claimed inventions were purportedly invented, renders the asserted claims invalid as obvious under 35 U.S.C. § 103.

The combination of references provided below and in the accompanying Prior Art Reference Charts in Exhibit A are exemplary and are not intended to be exhaustive. Additional obviousness combinations are possible, and BigCommerce reserves the right to use any such combination in this litigation. BigCommerce is unaware of Express Mobile's allegations with respect to the level of skill in the art and the qualification of the typical person of ordinary skill in the relevant art. BigCommerce is also unaware of the extent to which Express Mobile may contend that limitations of the claims are not disclosed in the prior art identified by BigCommerce as anticipatory, and the extent to which Express Mobile will contend that elements not disclosed in the asserted patent specification would have been known to persons of skill in the art. BigCommerce reserves the right to supplement these contentions to identify other references that would have made such limitations obvious in view of the relevant disclosures.

Patent Local Rule 3-3(b) requires an identification of a "motivation to combine" when a combination of times of prior art render an asserted claim obvious. In *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398 (2007), the Supreme Court held that prior art need not disclose the precise teachings of a patent to render it obvious because a court "can take account of the inferences and creative steps that a person of ordinary skill in the art would employ." *Id.* at 418. The Supreme Court rejected the Federal Circuit's prior rigid approach requiring "precise teachings directed to the specific subject matter of the challenged claim," and held instead that the obviousness analysis requires consideration of "ordinary skill and common sense." *Id.* at 481. As the Court

explained, “It is common sense that familiar items may have obvious uses beyond their primary purposes, and a person of ordinary skill often will be able to fit the teachings of multiple patents together like pieces of a puzzle.” *Id.* at 402. Under *KSR*, an explanation of why a combination of prior art items renders a claim obvious may be found in the “interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person of ordinary skill in the art ....” *Id.* at 418. The Supreme Court also rejected the view that the prior art references that are combined must address the same problem as the patented invention, stating that “any need or problem known in the field of endeavor at the time of the invention and addressed by the patent can provide reason for combining elements ... .” *Id.* The references listed above, alone or in combination, contain an explicit and/or implicit teaching, suggestion, motivation or inference to combine as noted in the relevant portion of the attached claim charts.

The Gever, Bernardo Ecommerce, Bernardo Help, Pinard, SilverStream, Miller, WebWriter I, WebWriter II, and Geocities prior art publications and system references are tightly clustered in the rather confined technological neighborhood of WYSIWYG browser-based HTML editors with effective dates before June 14, 1999, the earliest asserted applicable invention date. These references, notably, do not fall within the quite specific, in its own right, domain of website building tools, but actually fall within the even more specific sub-domain of browser-based WYSIWYG web publishing tools. This is the exact same confined space of the patents-in-suit. All these WYSIWYG browser-based HTML editors address the same specific problem—building websites without client-side software using just a browser. All WYSIWYG browser-based HTML editors offer the same solution—a build tool of some sort, an interface of some sort, some mechanism to intake certain data corresponding to website settings, some

mechanism to store that data, and finally some mechanism to query the database to read that data. They face the same technical and practical features, desirable functionalities, and technical challenges. For at least these reasons, a skilled artisan would have combined Gever, Bernardo Ecommerce, Bernardo Help, Pinard, SilverStream, Miller, WebWriter I, WebWriter II, and Geocities, where such combinations would achieve the various limitations relating to gathering data, previewing data, storing data, and building web pages. The claim chart contains exemplary excerpts for each of these references which, to a skilled artisan, would satisfy the corresponding claim element.

For independent claim 1 of both patents-in-suit and the balance of the independent claims for the '397 patents, each of the Gever, Bernardo Ecommerce, Bernardo Help, Pinard, SilverStream, Miller, WebWriter I, WebWriter II, and Geocities disclose content that a skilled artisan would interpret as satisfying every single claim element. This is not particularly surprising because the claims broadly attempt to privatize the building block steps of WYSIWYG browser-based HTML editors. *Any* attempt to gather data using a WYSIWYG browser-based HTML editor would require collecting data using a menu featuring selectable settings. *Any* attempt to gather data using a WYSIWYG browser-based HTML editor would involve data storage using a database and subsequent querying from the database (otherwise, what would the point of using the database be?). While the parties disagree about claim scope regarding various terms, under Plaintiff's broad constructions, *anything* evidently falls within the scope of a virtual machine including a browser's rendering engine; so *any* WYSIWYG browser-based HTML editor would feature a virtual machine and HTML commands would correspond to virtual machine commands. Thus, it is not surprising to find such a large number of references disclosing every single element of these independent claims.

To the extent *any* of those references' teachings is interpreted as falling short of any particular claim construction for any claim term in these claims, for the reasons explained in the preceding paragraph, there would be a trivially demonstrable motivation to combine teachings of any of the other references. For certain dependent claims requiring animations or transitions (cl. 8), buttons or images (cl. 9), transformations and timelines (cl. 11, '168 cls. 1 and 6), these are the building blocks of websites. Gever and Geocities, in particular, focus on animations, transitions and images. Thus Gever and Geocities, in combination with any of Bernardo Ecommerce, Bernardo Help, Pinard, SilverStream, Miller, WebWriter I, and WebWriter II, would render the above claims obvious. Other dependent claims recite multi-dimensional array databases (cl. 3), particular types of representative information (cl. 4), and particular types of elements (cls. 5, 6). Types of databases, information, and elements recited in these dependent claims were ubiquitous throughout the internet and the background art would have furnished explicit instances of each and every element of the Markush groups (of claims 4, 5, 6). Because these claims only require *one* of the listed Markush elements in the prior art, it would have been obvious to combine any reference not containing any of the Markush elements with any reference containing any *one* of the Markush elements.

### **III. DISCLOSURE UNDER PATENT RULE 3-3(c)**

The attached Exhibit A provides Prior Art Reference Charts for each item of prior art. The charts identify where specifically in each alleged item of prior art each element of each asserted claim is found. While each element of each asserted claim is found in each item of prior art in multiple locations, the attached charts provide examples of citations sufficient to identify at least one such location where each claim limitation is found in each item of prior art. Each item of prior art, however, discloses each claim limitation as a whole and specific citations are only

exemplary. Accordingly, BigCommerce reserves the right to rely on uncited portions of the prior art references as the citations must be interpreted in light of the entire disclosure of each reference. In addition, because persons of skill in the art generally would appreciate an item of prior art in context of other publications, literature, products, and understanding, BigCommerce reserves the right to rely on other publications and expert testimony as aids in understanding and interpreting the cited portions, for providing context to them, and as additional evidence that the prior art discloses a claimed feature. BigCommerce reserves the right to establish what was known to a person of having ordinary skill in the art through other publications, products, and/or testimony to establish that a person of skill in the art would have been motivated to combine the references rendering the claim obvious.

BigCommerce reserves the right to modify, amend, and/or change its interpretation of the prior art as additional or new constructions of the claim limitations may be provided by the Court, based on additional analysis by BigCommerce's technical expert witnesses, and/or based on other circumstances that may affect the meaning or application of the claims.

#### **IV. DISCLOSURE UNDER PATENT RULE 3-3(d)**

The asserted claims are invalid under 35 U.S.C. § 112.

The patent repeatedly casts the invention as a trifecta involving (1) HTML, (2) JavaScript, and (3) Java. Using Java aids bi-directional communication between the build engine (Java) and the interface (JavaScript+Html). That is the advance over the prior art from the vantage point of a skilled artisan deemed to read the intrinsic record cover to cover. Today, Express Mobile is revising the invention to an unrecognizable form, one neither enabled, nor described as Mr. Rempell's invention, nor possessed by Mr. Rempell at the time of the invention. Under Express Mobile's infringement theories, as conveyed in its infringement contentions and

the claim construction proposes it has submitted in unrelated litigation, it stakes claim to a different invention where a browser's generic rendering engine qualifies as a virtual machine. But the specification does not not enable an artisan to use a browser's rendering engine at the time of the invention. The written description does not evidence Mr. Rempell's possession of the invention as construed by Express Mobile. And the written description does not support the contention that Mr. Rempell regarded as his invention, what was subsequently claimed in the claims in either patent.

In the alternative, Defendant maintains that the asserted claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the terms "virtual machine," "commands to said virtual machine," "substantially contemporaneously," "information representative of . . . settings in a database," "generating a website," "retrieving said information," "building one or more web pages," "run time file," "portion of said database," "utilizes information," "portion of said one or more web pages," "generate virtual machine commands," "multi-dimensional array structured database," "vector object," "transition," "animation," "transformation or a timelines," "transformation," "timelines," "build engine," "accept user input," "associate a style," "values defining transformations and timelines," "transformations and timelines," "each web page is defined entirety by each of the plurality of objects comprising that web page and the style associated with that object," "object number," "runtime engine," "generate the web-site," and "extracted from the provided database" and therefore the scope of the invention claimed in the asserted patents.

## **V. DOCUMENT PRODUCTION**

Based on its investigation to date and under P.R. 3-4, BigCommerce has previously

produced Bates-stamped documents. In addition, BigCommerce's source code is currently available for inspection in accordance with the protective order entered in this case.

Dated: September 26, 2017

Respectfully submitted,

/s/ Amit Agarwal

Amit Agarwal  
California State Bar # 294269  
14420 Edinburgh Moor Dr.  
Wimauma, FL 33598  
Telephone: (310) 351-6596  
Email: [ama7386@gmail.com](mailto:ama7386@gmail.com)

Bobby Lamb  
State Bar No. 24080997  
GILLAM & SMITH, L.L.P.  
303 South Washington Avenue  
Marshall, Texas 75670  
Telephone: (903) 934-8450  
Facsimile: (903) 934-9257  
Email: [wrlamb@gillamsmithlaw.com](mailto:wrlamb@gillamsmithlaw.com)

*Attorneys for Defendant  
BigCommerce, Inc.*

**CERTIFICATE OF SERVICE**

I hereby certify that on September 26, 2017 I served the foregoing document to the attorneys of record via electronic mail.

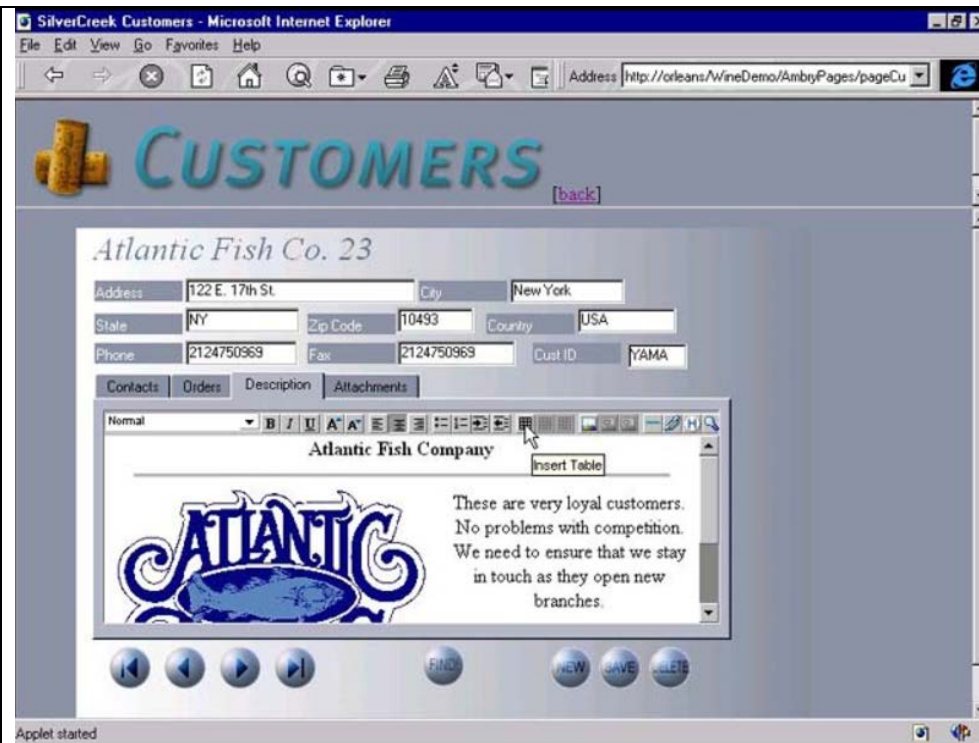
/s/ Bobby Lamb  
Bobby Lamb

# EXHIBIT A

Except where expressly noted otherwise, Defendant proffers these invalidity contentions based on the actual and apparent claim constructions taken by the plaintiff in its prior litigation and/or as reflected in its infringement contentions, but does not believe that these claim constructions are necessarily proper.

‘397 PATENT	
CLAIM 1	Invalidity Contentions
<p>A method to allow users to produce Internet websites on and for computers having a browser and a virtual machine capable of generating displays, said method comprising</p>	<p>Various prior art references teach a website building technology for computers using browsers and virtual machines and therefore discloses a method to allow users to produce Internet websites on and for computers having a browser and a virtual machine capable of generating displays.</p> <p>A partial list of exemplary excerpts from each reference is provided below. A skilled artisan would interpret these exemplary disclosures to satisfy this claim element.</p> <p><b>Silverstream:</b> “With SilverStream, you can write and deploy applications in either HTML or Java. You can use virtually any “current” Web browser to view HTML forms and pages served up by a SilverStream Server . . . In Release 1.0.1, SilverStream uses Microsoft’s Virtual Machine for Java for its standalone components: the Designer, SilverClient and Windows NT based SilverStream Server.” (SilverStream FAQ). This virtual machine for Java is precisely what is disclosed in the specifications.</p> <p><b>WebWriter II</b> “When the browser receives the HTML page generated by the server, it interprets the page by running the JavaScript function definitions, creating the JavaScript document tree and storing it at the top-level browser window. Then, it creates the frames and requests from the server the content of each frame, starting a “meteor shower” of HTML pages from the server to the browser, as shown in Figure 5.”</p> <p><b>WebWriter I</b> is “an integrated system for constructing Web applications that supports the creation of Web pages by non-programmers. WebWriter includes a direct manipulation Web page editor, the <i>WebWriter Editor</i>, which runs in a Web browser as CGI service, and the <i>Web Writer Page Generator</i>, which creates new pages as an applications runs.”</p> <p><b>Geocities Reference:</b> “In the fall of 1998, GeoCities launched an intuitive Web-page builder for with little technical expertise, especially with HTML. This WYSIWYG (What You see Is What You Get) editor provides you with authoring, site management, and image-editing tools without the need for HTML knowledge. With GeoBuilder, you can move items by drag-and-drop; resize your graphics and photos; establish links to other Websites; position multiple items simultaneously. Also included are an expanded clip-art library, numerous templates, and animated images. . . . GeoBuilder offers you design shortcuts to simplify your Web page-building process. . . . Click on a GeoBuilder link on the home page or go straight to <a href="http://www.geocities.com/members/tools/editor/inter.html">http://www.geocities.com/members/tools/editor/inter.html</a>.” “GeoBuilder works directly inside your Web browser, making it a breeze to use.”</p> <p><b>Gever:</b> “In preferred embodiments of the present invention, a server provides a Web site to which a user connects in order to create one or more individualized Web page components . . . [T]he present invention can . . . be adapted for building entire Web pages . . . Each of the animation sequences is generated by a respective script, preferably written in the JavaScript language, or as an executable Java program, which is stored on server 26. . . . The animation is carried out using a suitable Java player, which is downloaded implicitly to the computer of a visitor viewing the animation sequence.”</p> <p><b>Bernardo (Help):</b> “To create a web site, a Web site creator (the person using the tool to create a web site) is prompted by the tool through a series of views stored in the tool to select the features and options desired for the Web site. Based on these selections, the tool prompts the web site creator to supply data to populate fields of the templates determined by the tool to correspond to the selected features and options. Based on the identified templates and supplied data, the tool generates the customized Web site without the web site creator writing any HTML or other programming code.”</p> <p>“As schematically illustrated in the block diagram of FIG. 1, one aspect of the invention relates to a client/server network system <b>110</b> enabling access to both HTML objects <b>150 a-150 n</b> (e.g. HTML documents) and non-HTML objects <b>118 a-118 n</b> using a web browser <b>128</b> residing on a terminal <b>126</b>.</p>

	<p>Preferably, web browser <b>128</b> supports Internet and Web protocols (e.g., HyperText Transfer Protocol (HTTP) and TCP/IP). The system may further comprise a client <b>112</b> to enable a non-browser client access to non-HTML server module <b>124</b> to retrieve non-HTML objects <b>118a-118 n</b> from non-HTML database <b>116</b>.”</p> <p>“Thus, one advantage of the invention is that server <b>114</b> enables a browser <b>128</b> to request both HTML objects <b>150 a-150 n</b> and non-HTML objects <b>118a-118 n</b>. Additional advantages derive from this capability. For example, one significant advantage is that the invention exposes powerful resources available in non-HTML databases <b>116</b> to a web browser <b>128</b> according to one embodiment of the invention.”</p> <p><b>Bernardo (Ecommerce):</b> “To create a web site, a Web site creator (the person using the tool to create a web site) is prompted by the tool through a series of views stored in the tool to select the features and options desired for the Web site. Based on these selections, the tool prompts the web site creator to supply data to populate fields of the templates determined by the tool to correspond to the selected features and options. Based on the identified templates and supplied data, the tool generates the customized Web site without the web site creator writing any HTML or other programming code.”</p> <p>“Another object of the invention is to provide a tool for creating a Web site that minimizes or eliminates the need for a Web site creator to know or use HTML or other programming languages to create a Web site.”</p> <p><b>Pinard:</b> “The web page generator of the present invention provides for automatic web page creation of an organizational directory for use in an Internet and Intranet environment. The web page directory is created from data stored in the directory application component of the web page generator and, after being placed on a web server, allows any individual who has network access to the web server offering the web page to view the directory information in a web page format. In another aspect of the present invention, the web page generator generates the directory web pages dynamically on an as requested basis.”</p> <p><b>Miller:</b> “Turquoise is an intelligent browser and editor for the World Wide Web (WWW) that allows users to create dynamic pages by demonstration rather than by writing program code. With Turquoise, users without programming experience can create scripts that combine data from several Web pages, automate repetitive browsing or editing tasks, convert other data formats into Hypertext Markup Language (HTML), and process submitted forms . . . The Web browser/editor in the prototype system is AOLpress, which is freely available from America Online[1]. AOLpress is a WYSIWYG editor, enabling users to create and edit pages without learning HTML, including pages containing forms and tables . . . Turquoise is a WWW browsing and editing system that supports the creation of dynamic pages and personal assistants by demonstration.”</p>
<p>(a) presenting a viewable menu having a user selectable panel of settings describing elements on a website, said panel of settings being presented through a browser on a computer adapted to accept one or more of said selectable settings in said panel as inputs therefrom and where at least one of said user selectable settings in said panel corresponds to commands to said virtual machine</p>	<p>Various prior art references, given that they teach WYSIWYG web building on browsers, solicit or collect data using menus where the selections correspond to commands to the virtual machine as Express Mobile appears to understand that term. Each therefore discloses a method involving presenting a viewable menu having a user selectable panel of settings describing elements on a website, said panel of settings being presented through a browser on a computer adapted to accept one or more of said selectable settings in said panel as inputs therefrom and where at least one of said user selectable settings in said panel corresponds to commands to said virtual machine. A partial list of exemplary excerpts from each reference is provided below. A skilled artisan would interpret these exemplary disclosures to satisfy this claim element.</p> <p><b>SilverStream:</b></p> <p>“SilverStream's Java based HTML editor is a control that can be placed on any form. It is generally bound to a single HTML field in the database. It is fully programmable, as well as providing a familiar Word-like user interface. Virtually all of the features of HTML 3.2 are supported, including tables, images, backgrounds, links, etc. It is also possible to paste in HTML created in another editor, or to enter raw HTML for proprietary tags. For users who prefer to use another editor, the SilverStream Server supports the standard HTTP PUT and GET commands. These are able to GET an HTML page from a database field, and PUT it back without the editor being aware that is accessing a database instead of a file system on the server.” (SilverStream Demo).</p>



**WebWriter II Reference/System:** “The object properties frame contains commands that are specific to the currently selected object.” This frame is on the bottom right and contains a menu with a user selectable panel of settings describing elements on a website where one of said user selectable settings in said panel corresponds to commands to said virtual machine such as italics and bold.

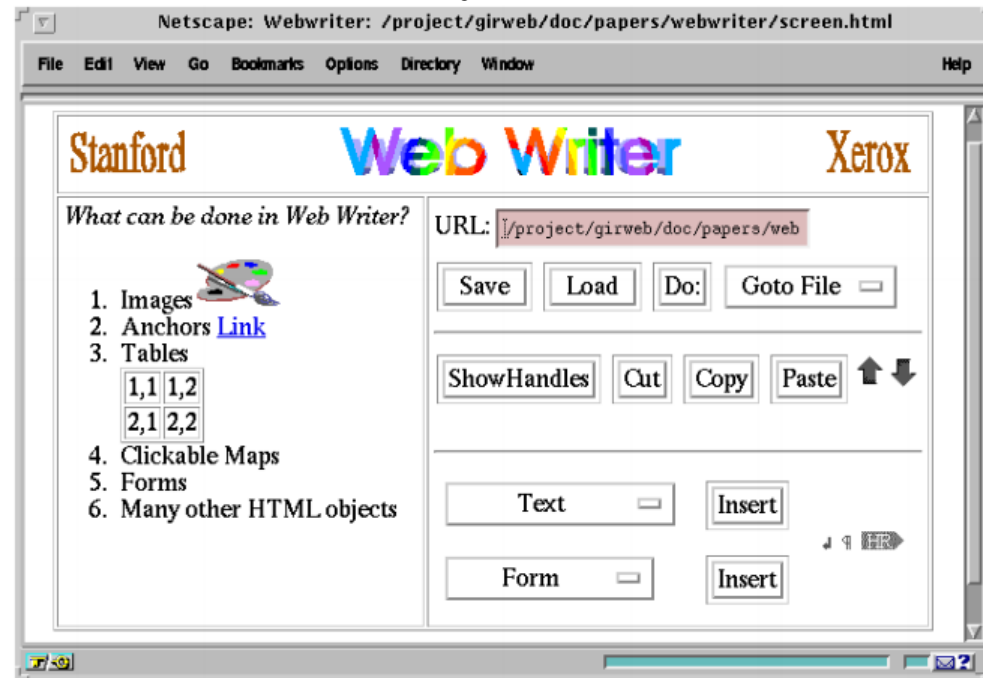


**Figure 1: The WebWriter II Editor.**

#### WebWriter I

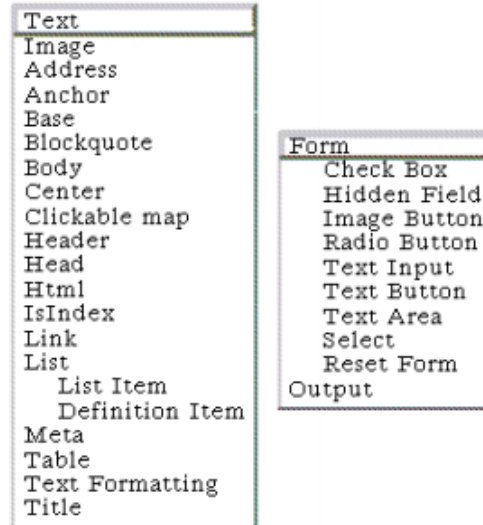
“To create template pages, the designer uses the WebWriter Editor, which is a general-purpose HTML editor that runs in a Web browser.” WW1 at 4.

“The editor has a two-column interface as shown in figure 2. ...



” WW1 at 4.

“To add content at any insertion point, the user clicks on that insertion point and then selects an HTML object type from either the Text of the Form menu, which are located as showing in Figure 2. ... The elements of these menus appear in figure 4.” WW1 at 7.



.. **Figure 4. The user can add any of the 32 HTML object types from these two menus.** ” WW1 at 7.

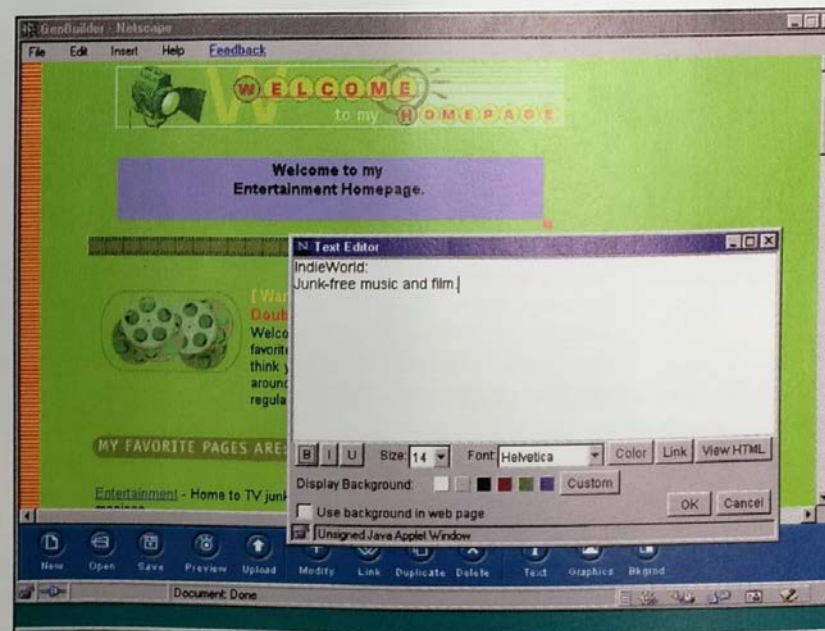
“Dynamic areas are inserted into a template page to indicate regions of the page that will be filled in at runtime by a program. There are two types of dynamic area supported by WebWriter: *variable areas* and *output areas*.” WW1 at 9.

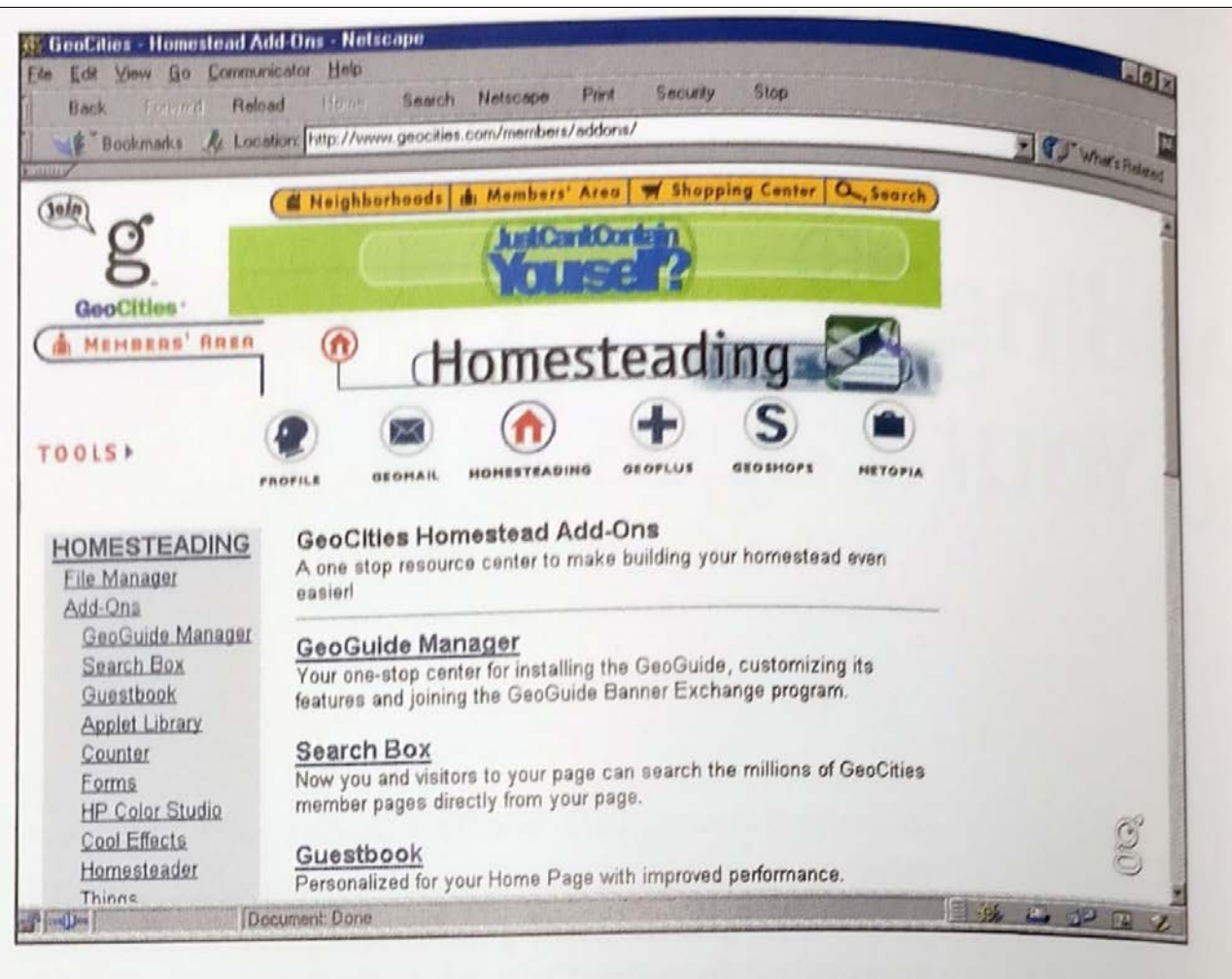
“The user creates an output area by selecting Output from the Form menu (recall figure 4) and clicking Insert. The output area object will be replaced at runtime by the output of a script, formatted as HTML and inserted at that point into the document.” WW1 at 9.

**Geocities:**

## Changing Words and Style

To change the heading, she double-clicked on the existing text, which brought up the Text Editor (Figure 6.3). You can also open the Text Editor by clicking the **Text** button.



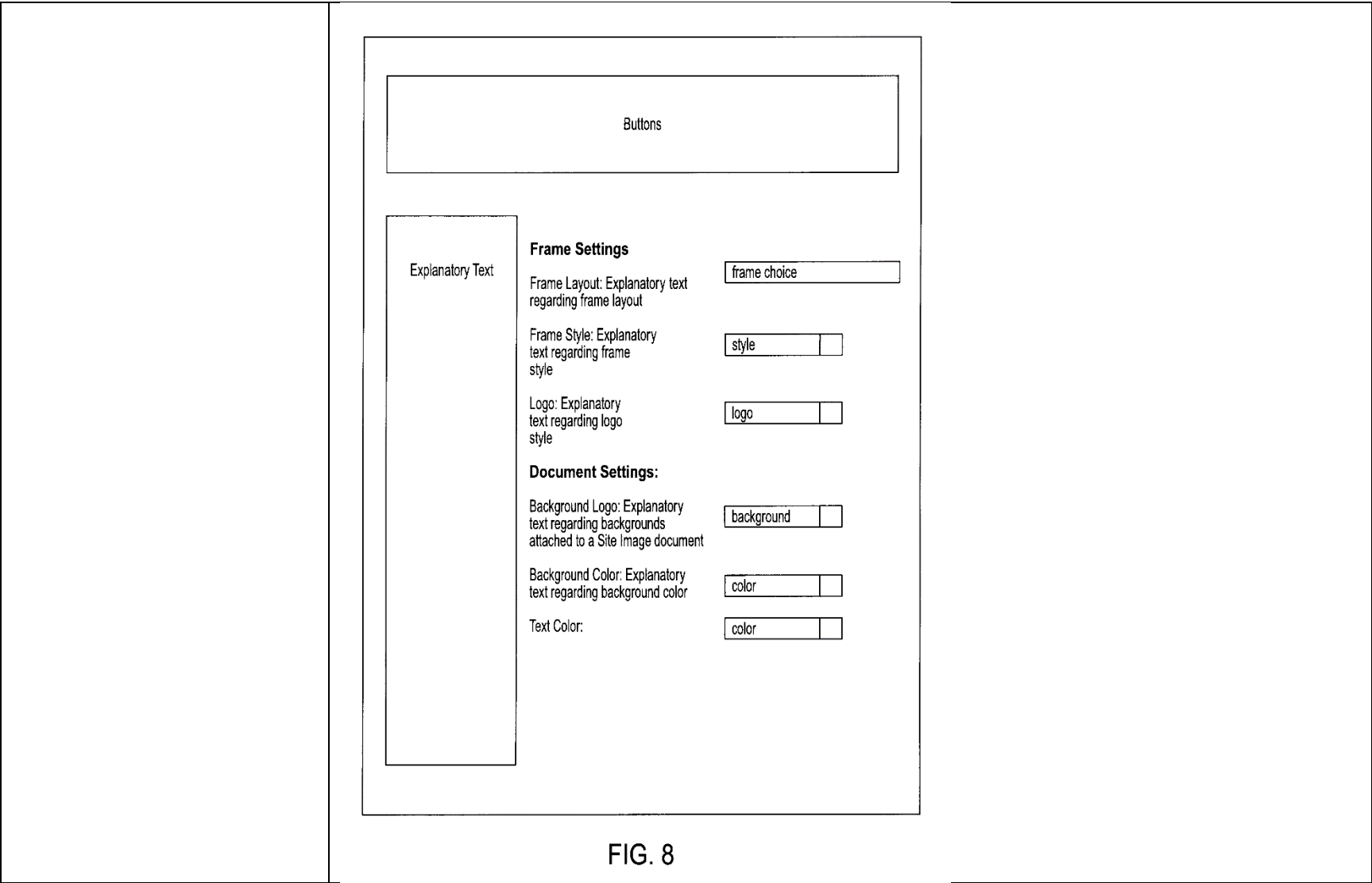


To install a GeoGuide, simply go to the GeoGuide Manager page within the Members' Area. Click on the **Set Up/Modify My GeoGuide** button. You'll then arrive at the GeoGuide Manager configuration page, which has a sample banner near the top of the page (see Figure 10.3). Below it is a list of the different parts of the banner from which you can choose the ones you want to appear on your page. You can have as many as nine different sections on your GeoGuide, in addition to the ad banner:

- ▶ Rank My Site—Visitors can rank your site for display on the GeoCities Avenues. GeoCities warns that this might not work on some framed sites.
- ▶ Take A Tour—Click this button to send your visitors on a guided tour of up to ten GeoCities pages that you choose.
- ▶ My Guestbook—A guestbook allows visitors to leave you messages, as well as comments and feedback about your site.
- ▶ Chat—This will send visitors to GeoCities chat, where they can interact with GeoCities Homesteaders and visitors.
- ▶ E-mail Me—Clicking on this button allows visitors to e-mail you directly at the e-mail address of your choice.
- ▶ Pages Like Mine—By clicking on this link, people can find pages that are similar to yours.
- ▶ Search—This link will send visitors to a GeoCities search page.
- ▶ Send This Page—If a visitor likes your site, he or she can click on this link and send a copy of your page to a friend.
- ▶ Forums—This links visitors to GeoCities' forums, where they can discuss a host of different topics.


**Gever:** FIG. 3 is a schematic view of an animation editing window 38 shown on display 22, in accordance with a preferred embodiment of the present invention. Editing window 38 comprises a display window 40 in which the selected animation sequence is displayed. A control panel 42 preferably comprises a play control 44; a freeze control 46, which allows a user to stop the sequence on a specific frame; and direction icons 48, which allow the user to scan the animation sequence. Alternatively or additionally, control panel 42 includes other suitable controls, such as a frame counter 49 and/or a scene counter 47. Window 38 preferably comprises a plurality of image attribute controls 52, which allow the user to change attributes of images shown in the animation sequence. The user preferably changes the attributes by selecting from a predefined group of replacements available from server 26. For example, in order to change a character 60 seen in the animation, the user selects the character, typically using a mouse or other pointing device, and actuates the appropriate control 52. A pop-up window 54 displays a plurality of characters from which the user may select a replacement for character 60. Preferably, the user is also able to change attributes of the selected character 56, such as the behavior of the character. Such behaviors are described, for example, in U.S. patent application Ser. No. 08/819,607, which is assigned to the assignee of the present patent application and is incorporated herein by reference, and in a corresponding PCT patent application published as WO97/35280. Likewise, the user may change border widths and colors of objects and of background regions in the animation sequence. Alternatively or additionally, window 38 allows changing of any other suitable attributes, such as the fonts and sizes of letters and background patterns.”

**Bernardo (Help):** “At step 6 a site creator is presented with one or more views. The view(s) preferably presents a screen containing a menu of features and/or options for creating and customizing the Web site. For example, FIG. 4 depicts an example of one view presented to a site creator. As shown, a view may include certain command initiators (e.g., buttons to click on), explanatory text fields to explain the objective or function of various aspects of the view and one or more feature(s) or option(s) to be selected. As shown, the user may be prompted to select whether to create a new site or edit an existing one.”



	<div>Buttons</div>
	<div><div>Explanatory Text</div><div>Make your selection, then click the "Next" button below. <input checked="" type="radio"/> Add Site Images <input type="radio"/> View/Update Site Images</div></div>

	<p><b>Bernardo (Ecommerce):</b> “According to another embodiment of the invention, the software tool provides a series of menus or views to guide the user through the creation of a web site, where the views comprise screens to enable the site creator to select the various features and options for the Web site and forms for entering text that is used to populate fields of stored templates.”</p> <p>“At step 6 a site creator is presented with one or more views. The view(s) preferably presents a screen containing a menu of features and/or options for creating and customizing the Web site. For example, FIG. 4 depicts an example of one view presented to a site creator. As shown, a view may include certain command initiators (e.g., buttons to click on), explanatory text fields to explain the objective or function of various aspects of the view and one or more feature(s) or option(s) to be selected.”</p> <p>“Another object of the invention is to provide a tool for facilitating the creation of Web sites and pages by taking a web site creator through a series of views, each having one or more options/features, to enable the site creator to select from a plurality of options/features available for the web site layout, content and functionality.”</p> <p><b>Pinard:</b> “When the appropriate downloadable platform independent application button is pressed by a local or remote user, the associated Java applet is then downloaded and run to communicate with the PBX 180 to determine the telephone status and features of that user's set, and to display those settings on the web page. Such settings can optionally be changed by a user with appropriate security clearance. Virtually any PBX feature, for example, call forwarding, call screening, or the call preferences for the person, can be added to an individual's web page, and easily generated by an administrator utilizing the automatic web page generator application of the present invention.”</p> <p><b>Miller:</b></p>
--	--



The screenshot shows a web browser window titled "Sam's Sandwiches". The page has a title "Sam's Sandwiches" and a form with the following fields:

- Name:** A text input field containing "Harry".
- Email address:** A text input field containing "harry".
- Clear Form:** A button located below the email address field.
- Sandwich:** A dropdown menu with four options: "Grilled Cheese", "Tuna Salad", "Turkey Club" (which is highlighted), and "Pita Pocket".
- Special requests:** A text input field containing "no mayonnaise".
- Order Lunch:** A button located at the bottom center of the form.

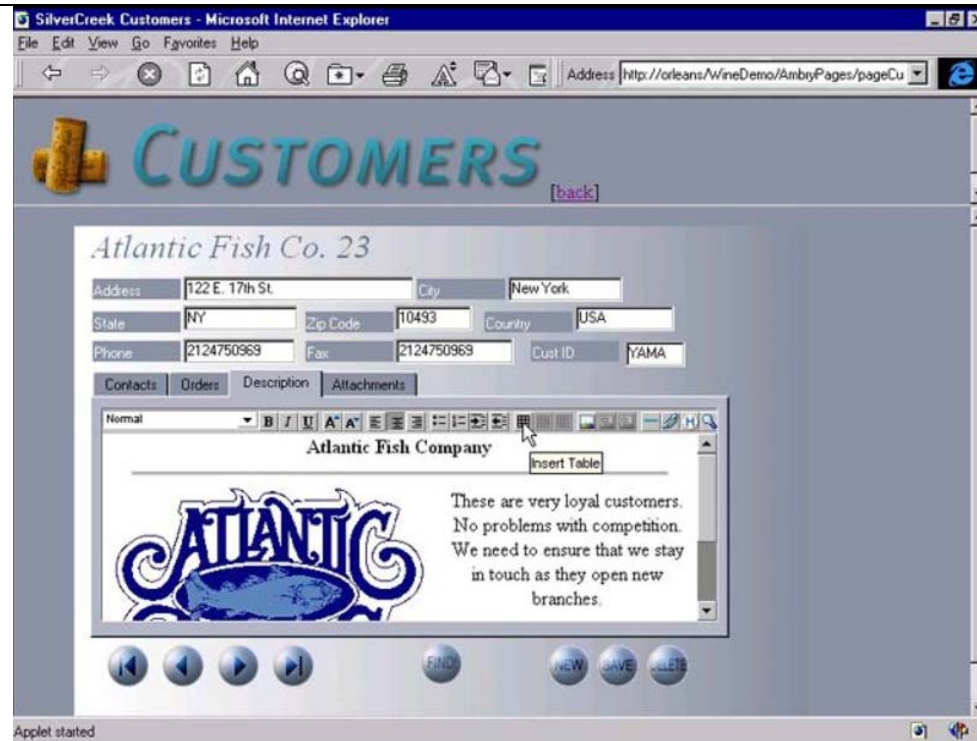
(a)



The screenshot shows a web browser window titled "Orders". The page has a title "Orders" and a table with three columns: "Sandwich", "Special Requests", and "Bought for".

Sandwich	Special Requests	Bought for
Turkey Club	no mayonnaise	Harry

	<p>To the extent this claim step is not disclosed in any one of the above references, it would have been obvious that reference <i>any</i> of the other references in which it is disclosed or to modify the reference under consideration accordingly. Based on the knowledge of a skilled artisan, this modification or combination would have been obvious because WYSIWYG interfaces using menus proliferated the background art, a WYSIWYG interface is the easiest way to intake inputs from non-technical users. For example, WebWriter I explicitly states that the prior art recognized the benefit of providing a website design solution “without the need to learn HTML” A WYSIWYG point-and-click menu for intaking user settings would advance this recognized need in the prior art.</p>
(b) generating a display in accordance with one or more user selected settings substantially contemporaneously with the selection thereof;	<p>Various prior art references in the WYSIWYG space must show users what they are doing as they select various options. That’s what makes the prior art WYSIWYG systems WYSIWYG. Thus, each of these references discloses a method involving generating a display in accordance with one or more user selected settings substantially contemporaneously with the selection thereof. A partial list of exemplary excerpts from each reference is provided below. A skilled artisan would interpret these exemplary disclosures to satisfy this claim element.</p> <p><b>See claim 1(a).</b></p> <p><b>SilverStream:</b> As shown below, the “Atlantic Fish Company” font appears Bold instantly after one clicks on the “B” setting below. This occurs substantially contemporaneously as confirmed by the reference’s explicit comparison to the “familiar Word-like user interface”</p>



#### WebWriter II Reference/System

The reference states, “After the user makes an edit to the document, the screen is redisplayed by calling the JavaScript reload() method on the preview frame. This updates the display without requiring any significant interaction with the server (because preview.html is cached at the browser and the document tree is converted to HTML as the browser interprets preview.html).”

#### WebWriter I

“In editing mode, WebWriter displays the current page as interpreted HTML together with additional objects, called *insertion points* and *handles*, that aid in the editing process.” WW1 at 4.

“[I]n WebWriter, the user builds an application as a sequence of Web pages, where each page can contain text, images, HTML forms, and content that is computed on the fly by WebWriter scripts.” WW1 at 2.



**Figure 3. Handles and insertion points are added to the document during editing.**

WW1 at 5.

“Insertions points, shown as diamonds in figure 3, indicate those points in the page where new content can be added.” WW1 at 5.

“To add content at any insertion point, the user clicks on that insertion point and then selects an HTML object type from either the Text of the Form menu, which are located as shown in figure 2.” WW1 at 7.

“We display the document and the control panel side by side in the WebWriter Editor.” WW1 at 16.

“The WebWriter Editor displays elements of Web page in a formatted style even during editing.” WW1 at 20.

Geocities:

### ► Font Color

You can also change the text color from basic black by clicking on **Color** and working with the Color Picker. The Color Picker lets you select from a wide variety of colors and then fine-tune them by sliding the three arrows back and forth. A Preview window shows you what you have done so far. Knowing she would eventually

## Intel.com Web Page Wizard

- ▶ Easy to use
- ▶ Provides four default looks
- ▶ Provides animation and video clips automatically
- ▶ Requires no HTML knowledge
- ▶ Limits direct HTML editing and customization

The Intel.com Web Page Wizard helps you build your page quickly even if you don't know any HTML. The tradeoff is that you must select from the default page styles provided by the Wizard. The interface is very intuitive and easy to use. You make choices in the left-hand frame, and the right-hand frame is automatically updated to show your changes. Follow these instructions to use the Web Page Wizard.

3. The Wizard guides you through nine separate steps, once the editor has been launched. Begin at Step 1, in the top left corner of your screen, by choosing a graphic style from the drop-down box. Continue on through the next eight steps in the Wizard, adding a title, text, and links to your Web page. Each step lets you add a set of information to your page, including a title, links, paragraphs of text, animation, layout preferences, and even video clips. As you work on your page, your changes are reflected on the right-hand side of the screen (Figure 4.3).

 Intel Web Page Wizard

**Gever:** The user may preview any of the animation sequences on display 22 in order to select an animation sequence which forms a basis of the Web page to be created for the user.

**Bernardo (Help):** "Another object of the invention is to provide a tool for creating a Web site where the tool comprises a library of stored templates (including fields) associated with different options/features for a Web site, and various help documents relating to the options/features. The tool prompts a user of the tool to select desired options/features from a list of possible options/features. Based upon the option/features selected, the tool

	<p>determines which of the stored templates (and fields) are to be used and the user is prompted to supply data to populate those fields. The tool uses the templates and user supplied data to create the web pages that make up a Web site. It also identifies the help documents that relate to the selected features/options and includes their help documents in the web site. The Web site may then be posted on a network, such as the Internet.”</p> <p>“According to one embodiment of the invention shown in FIG. 2, the tool may be installed on a server <b>30</b>. Installation of the tool on other types computer systems is also possible. Server <b>30</b> may be connected to various client terminals <b>34</b>. Typically, client terminal <b>34</b> may include a display <b>101</b> capable of presenting views <b>102</b>. Terminal <b>34</b> may also preferably include an input device <b>106</b>. The tool preferably is able to access a database <b>40</b> wherein a library of templates, views, and fields may be stored.”</p> <p>“Some embodiments of the invention may include a preview function. A preview function may enable the user to view the Web page as it is created. For example as various options are selected, a preview of the option(s) selected may be displayed for a user to observe. This ability is provided by the tool based on the templates without requiring the user to close the file and reopen it with a browser as would be the case if the user were writing HTML code. Thus, the tool enables pages/objects to be rendered on-the-fly.”</p> <p><b>Bernardo (Ecommerce):</b> “A preview function of some embodiments may allow the user creating content to preview the Web site as it would appear using various types of interfaces. Options 104 selected for the various user interfaces are preferably stored in a suitable storage device or database (e.g, database 40) for retrieval when appropriate.”</p> <p><b>Pinard:</b> It is also within the sphere and scope of the invention that the user could choose from a selection of templates as to the style of web page created. In an alternate embodiment (not shown), the directory application can be programmed to prompt the administrator to include a picture or logo for each entry, which can also be used in a specific instance of the web page. The web page template may incorporate additional properties that enhance the functions of the web page relating to features of an added class of item. Any valid HTML command, item, or aspect may be incorporated in the web page template. <i>See</i> Fig. 4a.</p>
--	---

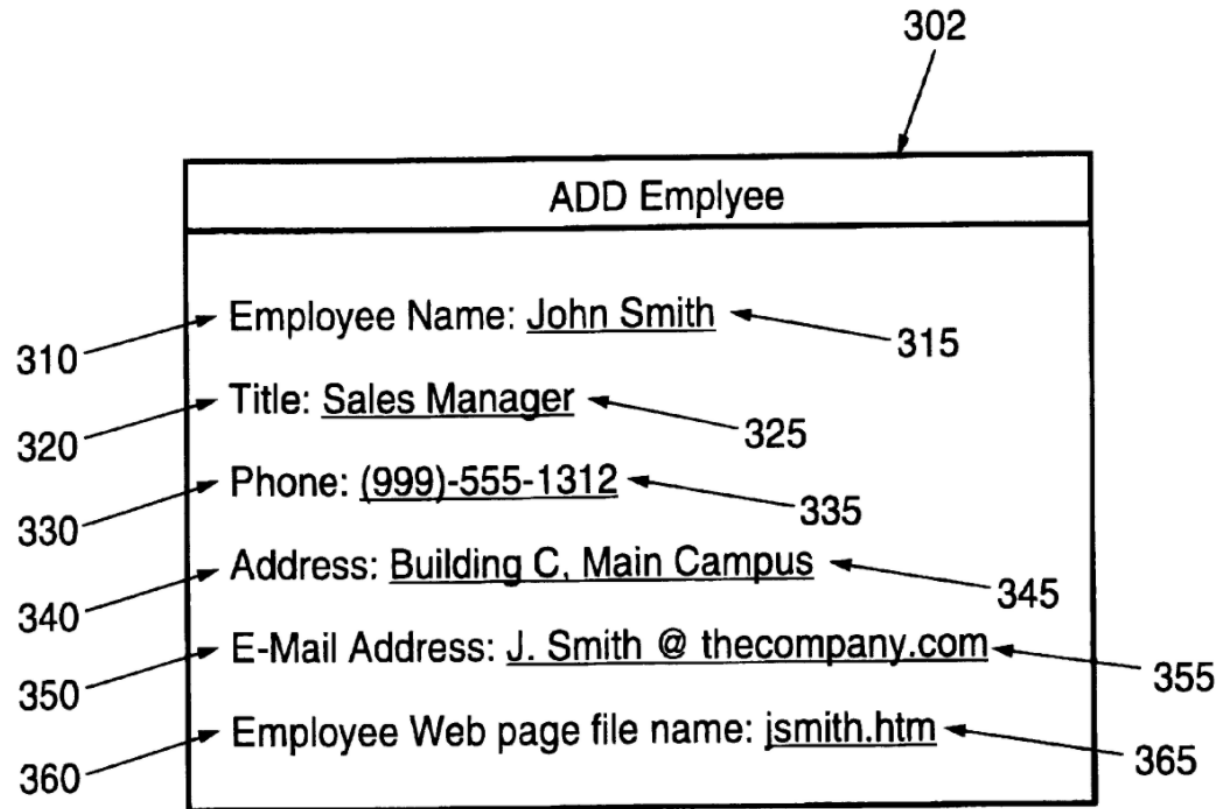


FIG. 4A

Miller: See 1(a).

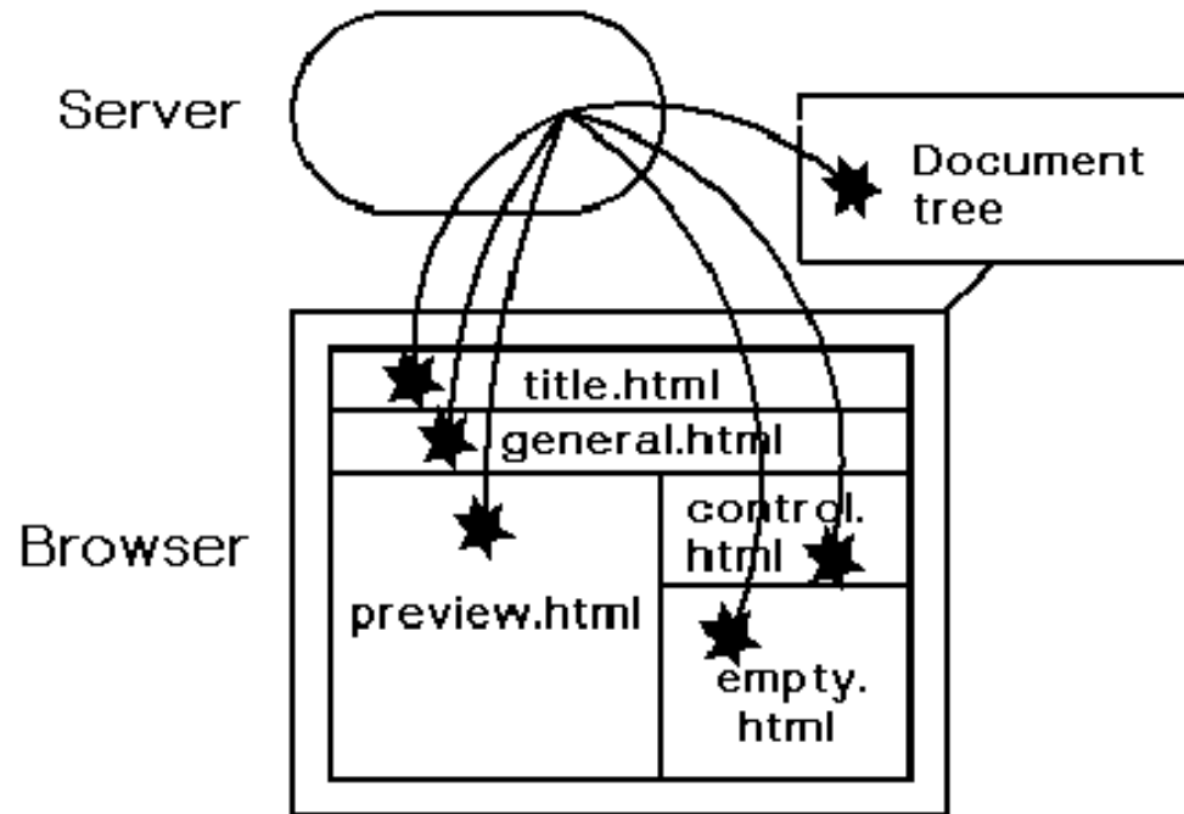
To the extent this claim step is not disclosed in any one of the above references, it would have been obvious that reference *any* of the other references in which it is disclosed or to modify the reference under consideration accordingly. Based on the knowledge of a skilled artisan, this modification or combination would have been obvious because WYSIWYG interfaces using menus proliferated the background art, a WYSIWYG interface is the easiest way to intake inputs from non-technical users. For example, WebWriter I explicitly states that the prior art recognized the benefit of providing a website design solution “without the need to learn HTML” A WYSIWYG point-and-click menu for intaking user settings would advance this recognized need in the prior art. Furthermore, WYSIWYG by definition requires previews. This motivation was recognized explicitly in virtually

	<p>every reference featuring WYSIWYGs. As just one example, consider WebWriter II, “Finally, for users who want to preview their HTML page in a particular browser, our implementation allows them to do this just by running WebWriter in the browser in question . . . placing the preview area in its own frame allows very long pages to be edited conveniently.”</p>
(c) storing information representative of said one or more selected settings in a database	<p>Various prior art references in the browser-based WYSIWYG space have to <i>store</i> the selections somewhere; not surprisingly, databases were a very common technique of storing data in the prior art. Various references disclose a method involving storing information representative of said one or more selected settings in a database. A partial list of exemplary excerpts from each reference is provided below. A skilled artisan would interpret these exemplary disclosures to satisfy this claim element.</p> <p><b>SilverStream:</b> “The SilverStream Server was designed to store all data in relational databases. This data includes structured relational data, application meta-data (forms, views, agents), content (HTML, and other content-related binary files), and security information. With all of this data in a database, you have the benefits of your DBMS, such as recovery, replication, and security features. In addition, you can take advantage of advanced SilverStream Server features such as versioning and full-text search capabilities of Web content.” SilverStream FAQ.</p> <p><b>WebWriter II Reference/System:</b> This reference implicitly discloses the storage of information representative of said one or more user selected settings in a database.</p> <p><b>WebWriter I</b></p> <p>“An user can save an individual page to disk by typing its file name into a field and clicking the Save button.” WW1 at 12.</p> <p>“Once a WebWriter applications is created and saved to dick, it can be run in one of two ways.” WW1 at 13.</p> <p>“The main structure encoded in the hidden tag is the content tree – a representation of the content of the page being edited and indexes into it that define the selected object and insertion point. . . . After executing each user-specified action, WebWriter generates the next page by traversing the content tree and asking all objects to print themselves in a format appropriate for the document area of the editor.” WW1 at 17-18.</p> <p><b>Geocities and Geocities S-1:</b></p> <div data-bbox="613 862 1453 1122" data-label="Text"> <p><b>GeoGuide</b></p> <p>The GeoGuide is a graphical banner that sits at the top of your page. You’ve probably seen these banners if you’ve visited other people’s GeoCities sites. The banner contains a number of options that visitors to your page may want to use, such as Send This Page, Rate My Site, Pages Like Mine, and Take a Tour. It also contains links to some other GeoCities–provided effects, like a guestbook and chat.</p> </div> <p>“Essentially, after you’ve configured your guide, GeoCities stores the profile in its internal database and you can update the settings any time.” Geocities. “User profile information is stored on multiple disk arrays using Informix Dynamic Server database software and backed up to long-term tape storage devices on a semi- hourly basis.” Geocities S-1 (1998).</p> <p><b>Gever:</b> There is additionally provided, in accordance with a preferred embodiment of the present invention, a method for creating an animation template for generating a computer animation, including . . . downloading a basic animation template from a server via a network, the basic template including one or more placeholders corresponding to respective fields in a database, such that an animation is generated responsive to the template including objects stored in the database;</p> <p><b>Bernardo (Help):</b> “The templates comprise databases which may include fields, forms, views, text, formulas and profiles that enable customization of the features. A profile may comprise fields, some of which may be designated as required fields.”</p>

	<p>“A server <b>114</b> preferably comprises a HTTP server module <b>130</b>, interface module <b>132</b> and non-HTML server module <b>124</b>. As further illustrated in FIG. 1, the server side of system <b>110</b> may comprise a non-HTML database <b>116</b> in communication with server <b>114</b>. Preferably, non-HTML database <b>116</b> stores one or more non-HTML objects <b>118 a-118 n</b>, each at least some of which having one or more non-HTML fields <b>162 a-162 n</b>, and a user directory <b>120</b>. User directory <b>120</b> includes one or more user objects <b>122 a-122 n</b>. User objects <b>122 a-122 n</b> may include information about a user (e.g., electronic mail address, role, profile, etc.). Terminal <b>126</b> and server <b>114</b> can be in communication via a network <b>154</b> (e.g., Internet, Intranet or other network).”</p> <p><b>Bernardo (Ecommerce):</b> “At step 20 the tool may populate template profile fields with the collected data. The database templates may preferably be constructed using an application assembly engine, for example, AppAssembler (see, FIG. 30). The database preferably stores the various templates, profiles, forms and views initiating the tool to finish the Web site may cause input data to populate the profiles and fields. ”</p> <p><b>Pinard:</b> “b) inputting member information into a directory database for each of a plurality of members;”</p> <p><b>Miller:</b> “Other systems give end-users the ability to develop CGI scripts, though not by demonstration. Several systems, like Zelig [15], represent a dynamic page as an HTML template with variable fields, which are computed at runtime by database queries or script code.”</p> <p>“Turquoise also draws ideas from p-strings [6], another language for searching structured-text databases.”</p> <p>To the extent this claim step is not disclosed in any one of the above references, it would have been obvious that reference <i>any</i> of the other references in which it is disclosed or to modify the reference under consideration accordingly. Based on the knowledge of a skilled artisan, this modification or combination would have been obvious in light of the explicit motivation recited in SilverStream, “With all of this data in a database, you have the benefits of your DBMS, such as recovery, replication, and security features.”</p>
(d) generating a website at least in part by retrieving said information representative of said one or more user selected settings stored in said database; and	<p>The <i>point</i> of WYSIWYG browser-based services in the prior art is to generate websites using the previously selected settings stored in the database. Thus, various prior art references disclose a method involving generating a website at least in part by retrieving said information representative of said one or more user selected settings stored in said database. A partial list of exemplary excerpts from each reference is provided below.</p> <p><b>SilverStream:</b> Below is depicted a website which is generated by retrieving information representative of user settings stored in the database. “When a user double-clicks on a wine in the list, SilverStream issues a command to the browser to display that wine’s Web page in the right hand frame. The SilverStream Server automatically maintains a URL for each wine and when a hit occurs on that URL, the server is able to retrieve the associated page from the HTML field in the database.”</p>



WebWriter II:



**Figure 5: The server startup meteor shower.**

“When the browser receives the HTML page generated by the server, it interprets the page by running the JavaScript function definitions, creating the JavaScript document tree and storing it at the top-level browser window. Then, it creates the frames and requests from the server the content of each frame, starting a “meteor shower” of HTML pages from the server to the browser, as shown in Figure 5.”

“Once a WebWriter applications is created and saved to dick, it can be run in one of two ways.” WW1 at 13.

“When a WebWriter application is running, each new page is assembled by the WebWriter Page Generator, another server-based CGI C++ program that knows how to read a template page, run the scripts specified in that template page and create a new page for display to the user.” WW1 at 18.

“The main structure encoded in the hidden tag is the content tree – a representation of the content of the page being edited and indexes into it that define the selected object and insertion point. ... After executing each user-specified action, WebWriter generates the next page by traversing the content tree and asking all objects to print themselves in a format appropriate for the document area of the editor.” WW1 at 17-18.

“The WebWriter Page Generator produces a new page based on a static page template and the output of one or more scripts. ... By packaging WebWriter with different scripting languages or built-in applications, the designer can strike a balance between a powerful system that can build many applications, as when the Unix shell interpreter is used, and one that is safe for a broader base of users, as when a simple set of forms-based applications is provided.” WW1 at 22.

**Geocities:**

**Site-Management Features**

The GeoPlus Manager is one of the site-management tools and is used as your Website control center. Here you can find account services, check out your most current Web statistics, or sign up for a Virtual URL and incremental space. You can also reach the Tools Manager inside the GeoPlus Manager. The Tools Manager gathers all the special tools and applications offered in the GeoPlus membership. You can access your File Manager to update your Website as well as all the utilities you'll need, including the Home Page Editors and EZ Upload.

**Gever:** “It is noted that although the above description refers to an animation sequence which is used as a component of a Web page, the present invention can also be adapted for building entire Web pages . . . There is additionally provided, in accordance with a preferred embodiment of the present invention, a method for creating an animation template for generating a computer animation, including downloading a basic animation template from a server via a network, the basic template including one or more placeholders corresponding to respective fields in a database, such that an animation is generated responsive to the template including objects stored in the database.”

**Bernardo (Help):** See 1b.

**Bernardo (Ecommerce):** “Each feature may have various options. To create a web site, a Web site creator (the person using the tool to create a web site) is prompted by the tool through a series of views stored in the tool to select the features and options desired for the Web site. Based on these selections, the tool prompts the web site creator to supply data to populate fields of the templates determined by the tool to correspond to the selected features and options. Based on the identified templates and supplied data, the tool generates the customized Web site without the web site creator writing any HTML or other programming code.”

**Pinard:** “c) retrieving said member web page template and said member information from said database;

d) for each of said plurality of members, replacing said fields in said member web page template with said member information and saving said member web page template with said fields replaced with said member information as a member specific web page;

e) retrieving said parent web page template;

f) for each member in said database, inserting said member information regarding each said member from said database into said fields of said parent web page template and creating links to associate each member whose member information has been inserted into said fields of said parent web page template to said member specific web page to create a new parent web page.”

	<p><b>Miller:</b> <i>See</i> 1a, 1c (showing the step of generating a website at least in part by retrieving said information representative of said one or more user selected settings stored in said database).</p> <p>To the extent this claim step is not disclosed in any one of the above references, it would have been obvious that reference <i>any</i> of the other references in which it is disclosed or to modify the reference under consideration accordingly. Based on the knowledge of a skilled artisan, this modification or combination would have been obvious in light of the explicit motivation recited in every single one of these references which recites building websites as the goal.</p>
<p>(e) building one or more webpages to generate said website from at least a portion of said database and at least one run time file, where said at least one run time file utilizes information stored in said database to generate virtual machine commands for the display of at least a portion of said one or more web pages</p>	<p>Again, the point of WYSIWYG browser-based editors is to build web pages from the information stored in the databases. That is the clear purpose of those databases. That data needs to be queried at some point subsequent to storing it. Otherwise, there is no purpose for storing it. Accordingly, various prior art references disclose a method involving building one or more webpages to generate said website from at least a portion of said database and at least one run time file, where said at least one run time file utilizes information stored in said database to generate virtual machine commands for the display of at least a portion of said one or more web pages. A partial list of exemplary excerpts from each reference is provided below. A skilled artisan would interpret these exemplary disclosures to satisfy this claim element.</p> <p><b>See claim 1 preamble.</b></p> <p><b>SilverStream:</b> “When a user double-clicks on a wine in the list, SilverStream issues a command to the browser to display that wine's Web page in the right hand frame. The SilverStream Server automatically maintains a URL for each wine and when a hit occurs on that URL, the server is able to retrieve the associated page from the HTML field in the database.”</p> <p><b>WebWriter II:</b></p> <p>“When the browser receives the HTML page generated by the server, it interprets the page by running the JavaScript function definitions, creating the JavaScript document tree and storing it at the top-level browser window. Then, it creates the frames and requests from the server the content of each frame, starting a "meteor shower" of HTML pages from the server to the browser, as shown in Figure 5.” <i>See generally</i> “Sec. 4.1.2 Loading and saving pages: Using the server to access files”</p> <p>The "Load" button is an example of an operation that requires the help of the server. We need to access the server for this operation because files are located in a server accessible file system. When the user clicks the "Load" button, the browser pops up a dialog box asking the user to supply the URL of a file to load. The information entered by the user is sent to a CGI script that runs on the web server. The server tries to read and parse the URL specified by the user. If this succeeds, the server sends a new frameset as in the startup process; but this time, instead of sending an empty document tree, it sends a representation of the document tree for the requested document. In fact, because the server is a Python program and the client is running JavaScript, the server encodes the document as a set of nested JavaScript calls to be interpreted by the client. These calls look like this: tree = new CreateChild(new CreateObject( new CreateState('h1', 'h1'), new CreateChild(new CreateObject( new CreateState('text', '', 'text', 'One Header', 'italics', 0, 'bold', 0, . 'One Header', null, null, 'text' )), "h1" )); where each CreateObject call adds a new object to the tree, and each object, in turn, may have children.”</p> <p>Sec. 4.1.3, entitled “Displaying the HTML page: Using the client to construct modified pages” incorporated herein by reference.</p> <p>“The architecture of the WebWriter II Editor -- multiple frames collaborating with one another via a browser scripting language and with the web server via CGI scripts -- can produce web applications that "in spirit" remain server-based, yet are highly responsive. This model we call the Meteor Shower Application Architecture, after the meteor shower of pages that the initial CGI script places into the browser's frames. The Meteor Shower Architectures relies in the functionality of the client browser and the server. The client browser provides processing for all interface operations, thereby ensuring short interactive response times. Since a Meteor Shower application is segmented into many frames, collaboration occurs among the frames as user operations in one frame cause other frames to update. Having many frames also limits updates to only those parts of the interface that require updating, potentially a significant savings because reload times can be relatively lengthy compared to the timeframe of individual interactions. Furthermore, the client can use in-place replacement of images to indicate state changes, wholly bypassing the need to reload or regenerate pages.”</p> <p><b>WebWriter I:</b></p>

"[T]he WebWriter system for non-programmers includes the *Web-Writer Page Generator*, a server-based CGI service that creates new pages as an application runs. Because they use the Page Generator, applications produced by WebWriter run as CGI programs on a Web server and hence can be used from many platforms and in many Web browsers."

"The user creates an output area by selecting Output from the Form menu (recall figure 4) and clicking Insert. The output area object will be replaced at runtime by the output of a script, formatted as HTML and inserted at that point into the document." WW1 at 9.

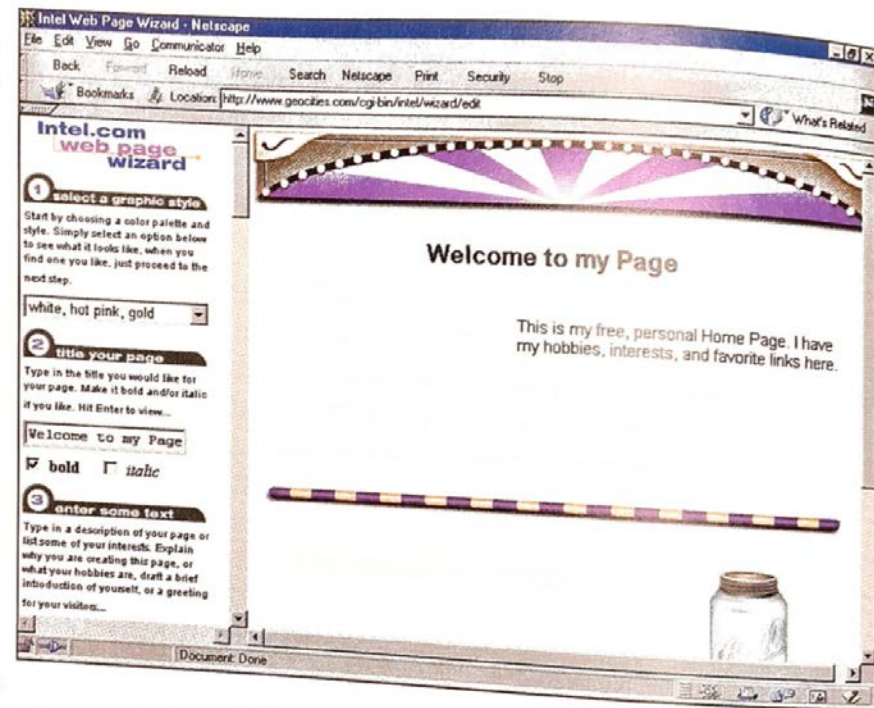
"The Page Generator generates a new web page on the fly as it scans through the template page." WW1 at 19.

"The WebWriter Page Generator produces a new page based on a static page template and the output of one or more scripts. ... By packaging WebWriter with different scripting languages or built-in applications, the designer can strike a balance between a powerful system that can build many applications, as when the Unix shell interpreter is used, and one that is safe for a broader base of users, as when a simple set of forms-based applications is provided." WW1 at 22.

**Geocities:**

screen (Figure 4.3).

**Figure 4.3**  
The Intel.com Web Page Wizard builds your Web page for you.



### **The CGI Library**

Do you want to boost your site's interactive capabilities? GeoPlus members have access to a special CGI Library that contains 12 CGI scripts. A *CGI script*, like the form handler, is a program that resides on the servers at GeoCities and gives you the ability to spice up your Web pages by offering new features, or processing information given to you by visitors to your page.

The CGI scripts included with GeoPlus allow you to build a guestbook, form automatic mailing lists, add a counter or clock, and include customizable surveys and forms. These CGI scripts allow you to better connect with the visitors to your site.

CGIs are self-contained programming modules that take data from your computer, run it into and through an application on the server, and return the information it collects to you.

A common reason GeoCities members upgrade to GeoPlus is that it offers a number of new CGI scripts which you can run on your page rather than just using the plain forms processor offered to basic GeoCities members.

These CGIs include:

- ▶ **Guestbook**—The Guestbook CGI enables members to leave messages on your site. It offers a lot more functionality than other guestbooks. For example, you can add customized imagery and backgrounds to the GeoPlus guestbook and edit posts.
- ▶ **Address Book**—The Address Book CGI lets visitors input and store their names and information in the address book. Think of it as a way for people to leave their contact information when they visit your site. The CGI also tells you the last CGI tool they used, such as whether they wrote you a note in the guestbook, left an answering machine message, or filled out a survey.
- ▶ **Answering Machine**—The Answering Machine CGI lets visitors to your site leave you notes, but unlike notes left in the guestbook, answering machine messages remain private and only you can view them.
- ▶ **Site Index**—This is an advanced CGI script for higher-end users. It lets you add a site index to your page. This enables people to search your pages for just the information they're looking for. You can place the script for the searching capabilities on every page. You'll have to place keywords on each page, then you can activate the indexing program so you have your own personal search engine for your site.
- ▶ **Clock, Countdown, and Countdown 2000**—These are three separate scripts that work in a similar fashion—all dealing with time. The Clock CGI lets you display the date and time on your page. You can set the clock to adjust for whatever time-zone you wish. The Countdown 2000 script displays how much time is left until the Year 2000. And the Countdown Clock CGI displays how much time is left until whatever date you might choose.
- ▶ **Counter**—Lets you add a page counter to any page on your site.
- ▶ **Simple Survey and Rank & Tell**—Lets you provide automatically updateable surveys to your site. You can build a survey that displays multiple-choice questions with a maximum of five answers.

Support for the CGI scripts is available to all GeoPlus members at [http://www.geocities.com/join/geoplus/bro\\_cgifaq.html](http://www.geocities.com/join/geoplus/bro_cgifaq.html)

### Additional Java Applets

GeoPlus also includes a bunch of special Java applets you can add to your page. These include:

- ▶ An animated logo applet that lets you show multiple graphics on one image area (similar to an animated GIF).
- ▶ A rotating sign applet that rotates an image within a banner-sized graphic, like an old gas station sign.

**Gever:** The animation is carried out using a suitable Java player, which is downloaded implicitly to the computer of a visitor viewing the animation sequence. Further details regarding implementation of such a Java player are described in the above-mentioned U.S. patent application Ser. No. 08/819,607 and in U.S. patent application Ser. No. 09/121,062, which are assigned to the assignee of the present patent application and incorporated herein by reference. In the example given in Table I, the user creates an on-line “photo album,” including animation effects, using editing window 38 or, most preferably, guided by the Wizard mentioned above.”

“Each of the animation sequences is generated by a respective script, preferably written in the JavaScript language, or as an executable Java program, which is stored on server 26 . . . The animation is carried out using a suitable Java player, which is downloaded implicitly to the computer of a visitor viewing the animation sequence.”

It is noted that although the above description refers to an animation sequence which is used as a component of a Web page, the present invention can also be adapted for building entire Web pages . . . There is additionally provided, in accordance with a preferred embodiment of the present invention, a method for creating an animation template for generating a computer animation, including downloading a basic animation template from a server via a network, the basic template including one or more placeholders corresponding to respective fields in a database, such that an animation is generated responsive to the template including objects stored in the database.

**Bernardo (Help):** “The tool has an application assembler module that combines the templates for the selected features/options with data input by the site creator to generate Web pages to thereby construct a complete Web site.”

“At step 20 the tool may populate template profile fields with the collected data. The database templates may preferably be constructed using an application assembly engine, for example, AppAssembler (see, FIG. 30). The database preferably stores the various templates, profiles, forms and views.”

**Bernardo (Ecommerce):** “A server 114 preferably comprises a HTTP server module 130, interface module 132 and non-HTML server module 124. As further illustrated in FIG. 1, the server side of system 110 may comprise a non-HTML database 116 in communication with server 114. Preferably, non-HTML database 116 stores one or more non-HTML objects 118 a-118 n, each at least some of which having one or more non-HTML fields 162 a-162 n, and a user directory 120. User directory 120 includes one or more user objects 122 a-122 n. ”

“At step 20 the tool may populate template profile fields with the collected data. The database templates may preferably be constructed using an application assembly engine, for example, AppAssembler (see, FIG. 30). The database preferably stores the various templates, profiles, forms and views initiating the tool to finish the Web site may cause input data to populate the profiles and fields. ”

**Pinard:** “In the preferred embodiment, the web server 110 supports the Common Gateway Interface (CGI) and is capable of running CGI programs. CGI is a publicly available method, used by web servers and web clients to mediate interaction between them.”

“In a further alternative embodiment, the template web pages discussed with respect to FIGS. 2-8 can incorporate applications that are embedded into the web pages generated by the method of the present invention by utilizing a downloadable platform independent application, such as a Java program. Java is a hardware independent interpreted language from Sun Microsystems® whose programs can be incorporated in HTML pages,

	<p>thereby enabling mini-programs called "applets" to be downloaded from a server, such as the web server 110 of FIG. 1, and run on client machines. This additional functionality can be incorporated into the web pages generated by the method of the present invention and utilized by a local user 160 or remote user 170, using a Java-enabled browser. The benefit of this alternative embodiment is that a sophisticated organizational directory may be maintained by an administrator without the necessity of the administrator having in-depth programming experience."</p> <p>"When the appropriate downloadable platform independent application button is pressed by a local or remote user, the associated Java applet is then downloaded and run to communicate with the PBX 180 to determine the telephone status and features of that user's set, and to display those settings on the web page. Such settings can optionally be changed by a user with appropriate security clearance. Virtually any PBX feature, for example, call forwarding, call screening, or the call preferences for the person, can be added to an individual's web page, and easily generated by an administrator utilizing the automatic web page generator application of the present invention."</p> <p><b>Miller:</b> "There is an alternative to traditional scripting languages: creating the script by demonstration. In programming-by-demonstration [11], the user describes a program by operating on example data, while the system watches and attempts to infer the intent of the user's actions. The end result of the demonstration is an abstract, executable program"</p> <p>To the extent this claim step is not disclosed in any one of the above references, it would have been obvious that reference <i>any</i> of the other references in which it is disclosed or to modify the reference under consideration accordingly. Based on the knowledge of a skilled artisan, this modification or combination would have been obvious in light of the explicit motivation recited in SilverStream about the advantage of using databases. "With all of this data in a database, you have the benefits of your DBMS, such as recovery, replication, and security features." There is no purpose of using databases if they are not subsequently used to build web pages.</p>
CLAIM 2	
An apparatus for producing Internet websites on and for computers having a browser and a virtual machine capable of generating displays, said apparatus comprising	<b>See claim 1 preamble.</b>
(a) an interface to present a viewable menu of a user selectable panel of settings describing elements on a website, said panel of settings being presented through a browser on a computer adapted to accept one or more of said selectable settings in said panel as inputs therefrom and where at least one of said user selectable settings in said panel corresponds to commands to said virtual machine	<b>See claim 1(a).</b>
(b) a browser to generate a display in accordance with one or more user selected settings contemporaneously with the selection thereof;	<b>See claim 1 (b).</b>

(c) a database for storing information representative of said one or more user selected settings;	<b>See claim 1(c).</b>
(d) a build tool having at least one run time file for generating one or more web pages, where said at least one run time file operating to utilize information stored in said database to generate commands to said virtual machine for generating the display of at least a portion of said one or more web pages	<b>See claims 1(d) and 1(e).</b>
CLAIM 3	
The apparatus of claim 2, wherein said database is a multi-dimensional array structured database	<p>A multi-dimensional array is an array that consists of other arrays</p> <ul style="list-style-type: none"> <li>● i.e., spreadsheet: <a href="http://www.homeandlearn.co.uk/java/multi-dimensional_arrays.html">http://www.homeandlearn.co.uk/java/multi-dimensional_arrays.html</a></li> <li>● i.e., <a href="https://www.w3schools.com/php/php_arrays_multi.asp">https://www.w3schools.com/php/php_arrays_multi.asp</a></li> </ul> <p>Various prior art references disclose use of a multi-dimensional array structured database. A partial list of exemplary excerpts from each reference is provided below. A skilled artisan would interpret these exemplary disclosures to satisfy this claim element.</p> <p><b>See claim1(c, d, e)</b> (disclosing use databases including multi-dimensional array structured databases for various types of data in every single reference). As just two examples, consider SilverStream and WebWriter I.</p> <p><b>SilverStream:</b> “The primary users of the SilverStream product will be corporations or Independent Software Vendors (ISVs) building Web applications in Java that need to access <i>relational databases</i> and rich content.”</p> <ul style="list-style-type: none"> <li>● SilverStream uses a relational database to store not only relational application data, but also <i>Web pages, content and all metadata such as forms, views and business logic</i>. This enables you to merge content with structured data like never before, while also providing easier deployment and better security. <ul style="list-style-type: none"> <li>○ This notion aligns with the spec’s description how arrays/databases are utilized (i.e., <i>see spec column 22, line 30</i>).</li> </ul> </li> <li>● “SilverStream Designer encompasses a complete set of visual design tools and wizards for Database Management: table definition (creates tables in the relational database) and table relationship definition.”</li> <li>● “A relational database organizes data in tables (or relations). A table is made up of rows and columns.” (i.e., <a href="https://www.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html">https://www.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html</a>)</li> </ul> <p><b>WebWriter I:</b></p> <p>“WebApp maintains state by asking every C++ object to save its relevant variables by pickling (representing as ASCII text) its data structures and saving them in a hidden field of a form.”</p>
CLAIM 4	
The apparatus of claim 3, wherein said representative information is Boolean data, numeric data, string data or multi-	Various prior art references feature Boolean data, numeric data, string data, or multi-dimensional arrays of various multimedia objects. A few illustrative examples are reproduced from 1(a) below:

dimensional arrays of various multimedia objects

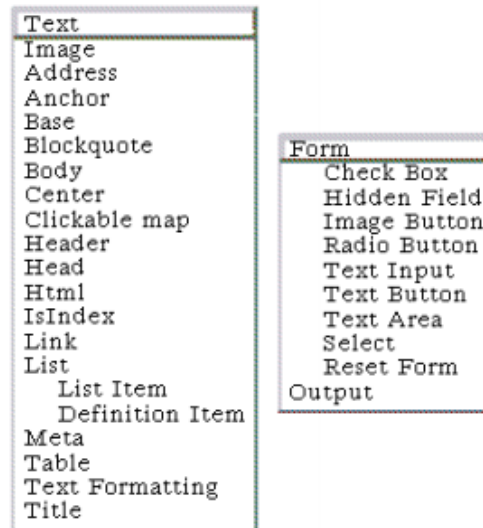
#### WebWriter I

"[I]n WebWriter, the user builds an application as a sequence of Web pages, where each page can contain text, images, HTML forms, and content that is computed on the fly by WebWriter scripts." WW1 at 2.

"To HideHandles, WebWriter changes the value of a Boolean variable to indicate that handles should be suppressed when each object generates its view in the new page; it then generates a new page without handles."

"The user may also specify a text string to show in place of the image in browsers that don't support images and may specify how to align the image relative to other objects on the same document line."

"The creation form for text allows specification of face (italics, bold, keyboard, or code)." WW1 at 7.



**Figure 4. The user can add any of the 32 HTML object types from these two menus.**

**Bernardo (Help):** "At step 8, a site creator may select the desired features/options. Selection may be accomplished in any suitable manner. For example, selection may be accomplished by using a graphic interface and selection device (e.g., by pointing a cursor at the selection and clicking on it), by entering text, or by other techniques. As shown in FIG. 5, desired site areas may be selected by entering an "X" in a check box. At step 10 it may be determined whether all selections have been made. If additional features/options are to be presented, the process returns to step 6 and repeats until all selections have been made and the process proceeds to step 12."

**Bernardo (Ecommerce):** "FIG. 8 illustrates another aspect of the web site development module. This design center feature provides a screen from which the user is prompted to decide on the options for the design of the web page, for example, by selecting colors, background images, frame styles, and logos. Once selections have been made, the user can preview the design and then make changes as desired."

**Pinard:**

The screenshot shows a web form titled "ADD Employee". It contains several text input fields, each with a label and a value. Numbered callouts point to specific parts of the form:

- 310 points to the "Employee Name:" label.
- 315 points to the value "John Smith" in the "Employee Name" field.
- 320 points to the "Title:" label.
- 325 points to the value "Sales Manager" in the "Title" field.
- 330 points to the "Phone:" label.
- 335 points to the value "(999)-555-1312" in the "Phone" field.
- 340 points to the "Address:" label.
- 345 points to the value "Building C. Main Campus" in the "Address" field.
- 350 points to the "E-Mail Address:" label.
- 355 points to the value "J. Smith @ thecompany.com" in the "E-Mail Address" field.
- 360 points to the "Employee Web page file name:" label.
- 365 points to the value "jsmith.htm" in the "Employee Web page file name" field.

**Gever:** "Pr  
texts, links,

les colors, characters, pictures,

**FIG.4A**

**Geocities:**

The object properties in GeoCities correspond to the representative information in the claim. For example, in the image below, the text editor contains Boolean data (whether boldface is toggled on or off), numeric data (the font size), string data (the text), or multi-dimensional arrays of various multimedia objects (the custom display background).

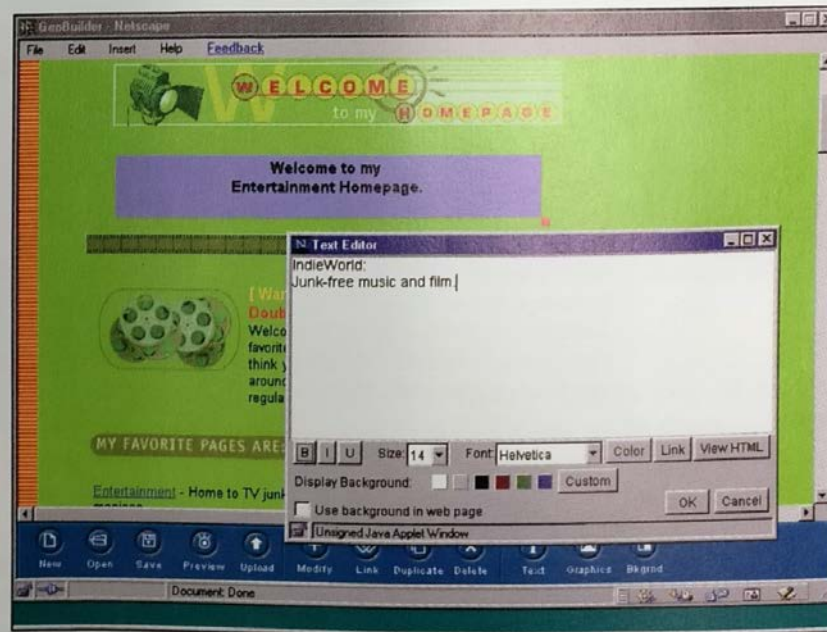
## Changing Object Properties

Each item you add to your page has certain properties that GeoBuilder assumes as a default. For example, all images are sized to their maximum size on your page, and text is left-aligned.

You can modify these default properties by selecting Edit, Properties from the GeoBuilder menu bar. Each object has its own set of properties that can be controlled, all of which are covered in-depth during this book as we come to a related topic.

## Changing Words and Style

To change the heading, she double-clicked on the existing text, which brought up the Text Editor (Figure 6.3). You can also open the Text Editor by clicking the **Text** button.

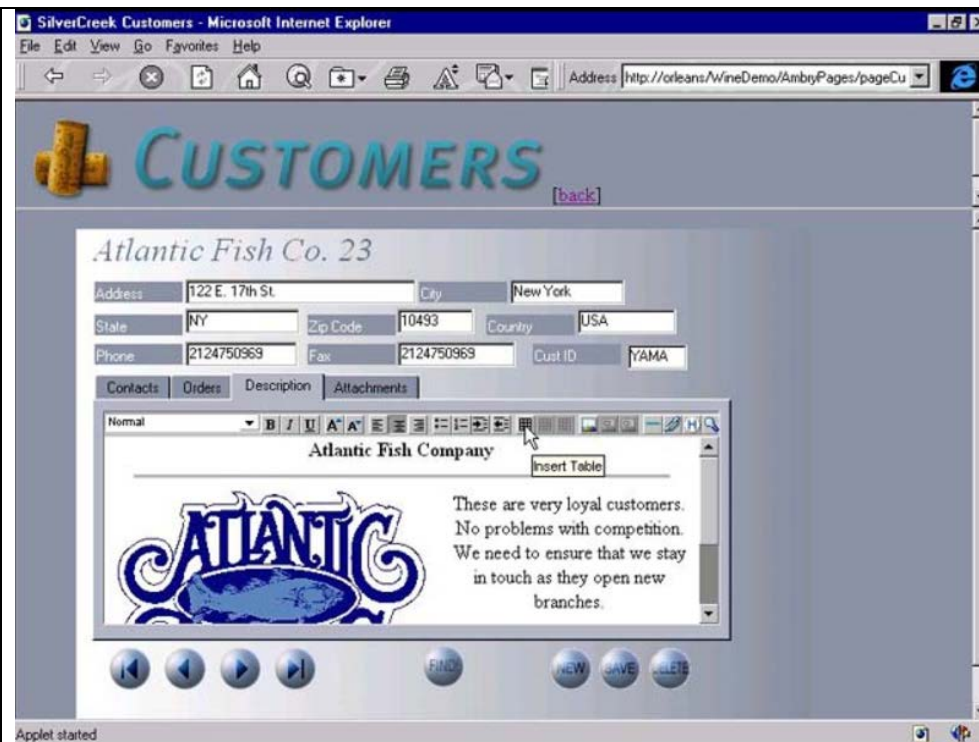


CLAIM 5

The apparatus of claim 4, wherein said elements include multimedia objects selected from the group consisting of a color, a font, an image, an audio clip, a video clip, a text area and a URL

Various prior art references disclose elements such as a color, a font, an image, an audio clip, a video clip, a text area and a URL. A partial list of exemplary excerpts from each reference is provided below.

**SilverStream:**



WebWriter II:

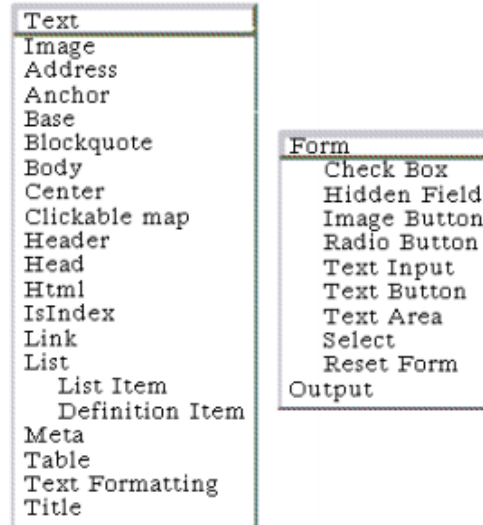


**Figure 1: The WebWriter II Editor.**

**WebWriter I:**

“[I]n WebWriter, the user builds an application as a sequence of Web pages, where each page can contain text, images, HTML forms, and content that is computed on the fly by WebWriter scripts.” WW1 at 2.

“The creation form for text allows specification of face (italics, bold, keyboard, or code).” WW1 at 7.



**Figure 4. The user can add any of the 32 HTML object types from these two menus.**

**Bernardo (Help):** “Content may include various objects and object types including, for example, text, graphics, other media and multimedia material such as audio and video presentations, and links to other objects including other Web pages. ”

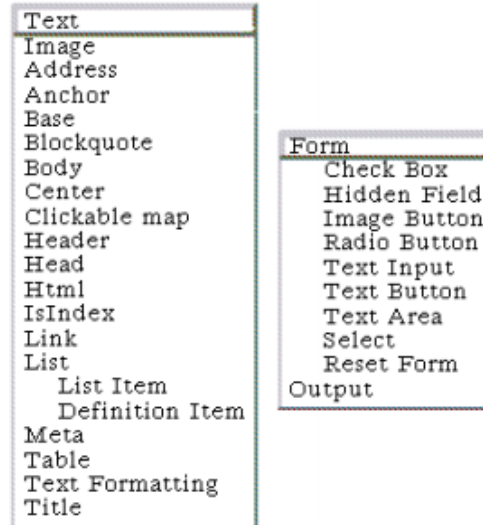
“The view may be any type of graphical or text arrangement that provides a user with several options. For example, view may include a drop down or pop up menu, a list of options to be selected using a radio button selector, a text entry box, or other suitable graphical selection interface. Options present a user with various choices regarding the creation of a Web site and can be based on stored templates. For example, options may include choices regarding editing of text, colors, graphics or other objects, as well as, choices regarding positioning of objects, creation of new objects, deleting objects, adding links to other sites, security provisions, and other choices. For example, FIG. 10 depicts a view that enables a content composer to add site images or to view/update site images. As illustrated the content composer selected the “add site images.””

“The features/options may pertain to those found on Web sites or other features/options. For example, the features/options may include a list of site areas, Web pages to include in a Web site, Web page formatting options, security links, colors, borders, buttons, workflow commands, graphics, backgrounds, text and other items. Some features or options may be required and some may be discretionary.”

**Bernardo (Ecommerce):** “Another object of the invention is to provide a tool for facilitating the creation of Web pages with templates for predefined Web pages that enables personalization and customization of the Web pages without the need for the user to change or write any software code, and facilitates the inclusion or modification of graphical and other multimedia objects.”

**Pinard:** “2. The method of claim 1 wherein said member information includes a reference for locating a digital picture of each said member, said digital picture being displayed when said member specific web page is accessed.”

	<p>“f) for each member in said database, inserting said member information regarding each said member from said database into said fields of said parent web page template and creating links to associate each member whose member information has been inserted into said fields of said parent web page template to said member specific web page to create a new parent web page;”</p> <p><b>Gever:</b> “The categories may include, for example, photo albums, advertisements, greeting cards, and resumes. Alternatively or additionally, the categories may be defined according to any other classification. According to the selected category, the user is provided with a list of titles of basic animation sequences. Each of the animation sequences is generated by a respective script, preferably written in the JavaScript language, or as an executable Java program, which is stored on server 26. ”</p> <p><b>Geocities:</b> “Multimedia Things—Specializes in helping you add multimedia things to your site, such as audio and video clips.”</p>
CLAIM 6	
<p>The apparatus of claim 2, wherein said elements are selected from the group consisting of a button, an image, a paragraph, a frame, a table, a form and a vector object</p>	<p>Various prior art references elements selected from the group consisting of a button, an image, a paragraph, a frame, a table, a form and a vector object. A partial list of exemplary excerpts from each reference is provided below.</p> <p><b>See claim 1(a)</b> which shows elements selected from a button, an image, a para, a frame, a table, a form, and a vector object. Consider, as just two examples:</p> <p><b>WebWriter I</b></p> <p>“[I]n WebWriter, the user builds an application as a sequence of Web pages, where each page can contain text, images, HTML forms, and content that is computed on the fly by WebWriter scripts.” WW1 at 2.</p> <p>“The image creation for a form allows the URL of the image to be selected from a menu or typed.” WW1 at 7.</p> <p>“By default, each page of a WebWrite application is an HTML form. As a form, each page can contain user interface elements, such as buttons, check boxes, radio buttons, and type-in fields.” WW1 at 9.</p>



**Figure 4. The user can add any of the 32 HTML object types from these two menus.**

**Geocities:** GeoCities enabled use of animation on websites: “Animated GIFs are a special version of the GIF format that allows you to create fun animations by embedding the same graphic files multiple images that are cycled through when loaded to create the appearance of animation—just like an old-fashioned flip book.”

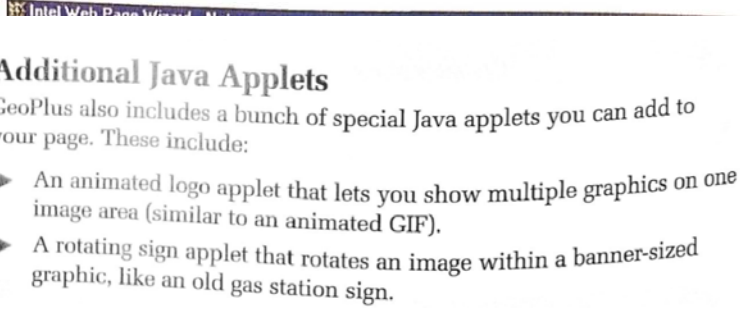
#### CLAIM 8

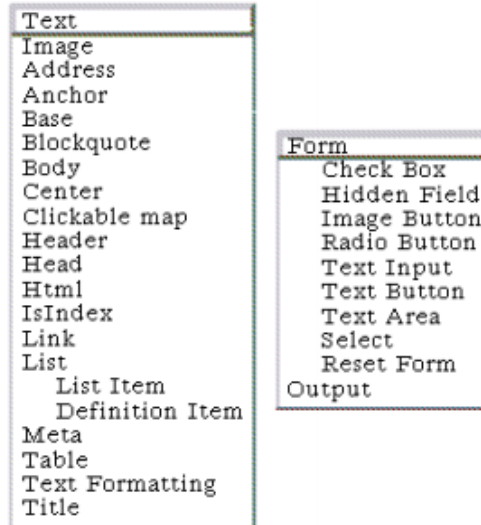
The apparatus of claim 2, wherein said elements include one or more objects on a web page, and wherein said description of elements are a transition or an animation of at least one of said elements on a web page

Various prior art references feature elements including one or more objects on a web page, wherein said description of elements are a transition or an animation of at least one of said elements on a web page. A partial list of exemplary excerpts from each reference is provided below.

**Gever:** FIG. 3 is a schematic view of an animation editing window 38 shown on display 22, in accordance with a preferred embodiment of the present invention. Editing window 38 comprises a display window 40 in which the selected animation sequence is displayed. A control panel 42 preferably comprises a play control 44; a freeze control 46, which allows a user to stop the sequence on a specific frame; and direction icons 48, which allow the user to scan the animation sequence. Alternatively or additionally, control panel 42 includes other suitable controls, such as a frame counter 49 and/or a scene counter 47. Window 38 preferably comprises a plurality of image attribute controls 52, which allow the user to change attributes of images shown in the animation sequence. The user preferably changes the attributes by selecting from a predefined group of replacements available from server 26. For example, in order to change a character 60 seen in the animation, the user selects the character, typically using a mouse or other pointing device, and actuates the appropriate control 52. A pop-up window 54 displays a plurality of characters from which the user may select a replacement for character 60. Preferably, the user is also able to change attributes of the selected character 56, such as the behavior of the character. Such behaviors are described, for example, in U.S. patent application Ser. No. 08/819,607, which is assigned to the assignee of the present patent application and is incorporated herein by reference, and in a corresponding PCT patent application published as WO97/35280. Likewise, the user may change border widths and colors of objects and of background regions in the animation sequence. Alternatively or additionally, window 38 allows changing of any other suitable attributes, such as the fonts and sizes of letters and background patterns.”

**Geocities:**

	<p>3. The Wizard guides you through nine separate steps, once the editor has been launched. Begin at Step 1, in the top left corner of your screen, by choosing a graphic style from the drop-down box. Continue on through the next eight steps in the Wizard, adding a title, text, and links to your Web page. Each step lets you add a set of information to your page, including a title, links, paragraphs of text, animation, layout preferences, and even video clips. As you work on your page, your changes are reflected on the right-hand side of the screen (Figure 4.3).</p>  <p><b>Additional Java Applets</b></p> <p>GeoPlus also includes a bunch of special Java applets you can add to your page. These include:</p> <ul style="list-style-type: none"> <li>▶ An animated logo applet that lets you show multiple graphics on one image area (similar to an animated GIF).</li> <li>▶ A rotating sign applet that rotates an image within a banner-sized graphic, like an old gas station sign.</li> </ul> <p>The above are two examples of elements including one or more objects on a web page, wherein the description of elements is a transition or an animation of at least one of said elements on a web page. Adding transitions or animations of elements was therefore commonplace in the prior art, one of the building blocks of webpages, and was inherently disclosed and/or obvious to implement in any of the prior art references listed in 1(a)-(e).</p>
CLAIM 9	
<p>The apparatus of claim 2, wherein said elements include a button or an images, wherein said selectable settings includes the selection of an element style, and wherein said build engine includes means for storing information representative of selected style in said database</p>	<p>See claims 1(a, c), 2(a, c), 3, 4, 5, 6, 8 above. Each of those contentions lists exemplary excerpts from the prior art featuring elements include a button or an images, wherein said selectable settings includes the selection of an element style, and wherein said build engine includes means for storing information representative of selected style in said database.</p> <p>As just one example, consider:</p> <p>“[I]n WebWriter, the user builds an application as a sequence of Web pages, where each page can contain text, images, HTML forms, and content that is computed on the fly by WebWriter scripts.” WW1 at 2. “The creation form for text allows specification of face (italics, bold, keyboard, or code).” WW1 at 7.</p>



**Figure 4. The user can add any of the 32 HTML object types from these two menus.**

CLAIM 11	
The apparatus of claim 9, wherein said elements are described by a transformation of a timelines of said selected styles	See claim 8.
CLAIM 37	
<p>An apparatus for producing Internet websites having one or more web pages on and for a computer having a browser and a virtual machine capable of generating a display, said apparatus comprising:</p> <p>(a) an interface configured for building a website through control of website elements, said interface being operable through the browser on the computer to:</p> <p>present a viewable menu of a user selectable panel of settings, accept a plurality of</p>	<p>See claims 1-11 (charted above). It would have been obvious to a skilled artisan to use internal and external databases. This particular configuration of databases does not add a non-obvious character to these claims because it is a non-obvious configuration to accomplish the same storage function.</p>

<p>settings from said user selectable panel of settings to form an assembly of settings, and</p> <p>generate the display in accordance with said assembly of settings contemporaneously with the acceptance thereof, at least one of said user selectable settings of said panel of settings being operable to generate said display through commands to said virtual machine;</p> <p>(b) an internal database associated with said interface for storing information representative of one or more of said assembly of settings for controlling elements of the website; and</p> <p>(c) a build tool to construct one or more web pages of said website having:</p> <p>an external database containing data corresponding to said information stored in said internal database, and</p> <p>one or more run time files,</p> <p>where said run time files utilize information stored in said external database to generate virtual machine commands for the display of at least a portion of said one or more web pages.</p>	
---	--

'168 PATENT: WEBWRITER 1	
A system for assembling a web site comprising: a server comprising a build engine configured to:	<i>See US6546397, claim 1-preamble.</i>
accept user input to create a website, the website comprising a plurality of web pages, each web page comprising a plurality of objects	<i>See US6546397, claims 1(a), 1(b), 1(c).</i>
accept user input to associate a style with objects of the plurality of webpages, wherein each web page comprises at least one button object or at least one image object, wherein the at least one button object or at least one image object is associated with a style that includes values defining transformations and timelines for the at least one button object or at least one image object; and wherein each web page is defined entirely by each of the plurality of objects comprising that web page and the style associated with that object.	<i>See US6546397, claim 1(a),(b),(c), 3, 4, 5, 6, 8, 11.</i>
Produce a database with a multidimensional array comprising objects that comprise the website including data defining, for each object, the object style, an object number, and an indication of the web page that each object is a part of, and	<i>See US6546397, claim 1(c), 3, 8, 11.</i>

<p>Provide the database to a server accessible to web browser</p>	<p><i>Id.</i></p>
<p>Wherein the database is produced such that a web browser with access to a runtime engine is configured to generate the web-site from the objects and style data extracted from the provided database.</p>	<p><i>See</i> US6546397, claim 1(e).</p>

CLAIM 6	
The system of claim 1, where said data is stored as one or more of a Boolean an integer, a string, a floating point variables, or a URL.	<i>See US6546397, claim 4.</i>

**Enablement or written description under 35 U.S.C. § 112(1)**

The patent repeatedly casts the invention as a trifecta involving (1) HTML, (2) JavaScript, and (3) Java. Using Java aids bi-directional communication between the build engine (Java) and the interface (JavaScript+Html). That is the advance over the prior art from the vantage point of a skilled artisan deemed to read the intrinsic record cover to cover. Today, Express Mobile is revising the invention to an unrecognizable form, one neither enabled, nor described as Mr. Rempell's invention, nor possessed by Mr. Rempell at the time of the invention. Under Express Mobile's infringement theories, as conveyed in its infringement contentions and the claim construction proposes it has submitted in unrelated litigation, it stakes claim to a *different* invention where a browser's generic rendering engine qualifies as a virtual machine. But the specification does not enable an artisan to use a browser's rendering engine at the time of the invention. The written description does not evidence Mr. Rempell's possession of the invention as construed by Express Mobile. And the written description does not support the contention that Mr. Rempell regarded as his invention, what was subsequently claimed in the claims in either patent.

**Indefiniteness under 35 U.S.C. § 112(2)**

In the alternative, Defendant maintains that the asserted claims, read in light of the specification delineating the patent, and the prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the terms "virtual machine," "commands to said virtual machine," "substantially contemporaneously," "information representative of . . . settings in a database," "generating a website," "retrieving said information," "building one or more web pages," "run time file," "portion of said database," "utilizes information," "portion of said one or more web pages," "generate virtual machine commands," "multi-dimensional array structured database," "vector object," "transition," "animation," "transformation or a timelines," "transformation," "timelines," "build engine," "accept user input," "associate a style," "values defining transformations and timelines," "transformations and timelines," "each web page is defined entirety by each of the plurality of objects comprising that web page and the style associated with that object," "object number," "runtime engine," "generate the web-site," and "extracted from the provided database" and therefore the scope of the invention claimed in the asserted patents.